INTEGRAL Science Data Centre

IBIS Analysis User Manual

Reference	:	ISDC/OSA-UM-IBIS
Issue	:	10.1
Date	:	3 September 2014

INTEGRAL Science Data Centre
Chemin d'Écogia 16
CH–1290 Versoix
Switzerland

http://isdc.unige.ch

Authors and Approvals

ISDC	IBIS Analysis User Manual	
3 September 2014	10.1	

Prepared by : M. Chernyakova
A.Neronov, L. Pavan & M. Türler

Agreed by : R. Walter ....

Approved by : T. Courvoisier ....

Document Status Sheet

ISDC	IBIS Analysis User Manual	
2 April 2003	1.0	First Release.
19 May 2003	1.1	Update of the First Release. Section 6, Tables 63, 10, 12, 13, 16, 52, 58 and Figures 10, 27 were updated. Section 12.9.1 was added.
18 July 2003	2.0	Second Release. Sections 5, 6, 8 and the bibliography were updated. Sections 12.9, 12.12.2, C.8, C.9.2 were added.
5 December 2003	3.0	Third Release. Sections 6, 7 and 8 were updated. Sections 9.11, C.9.3 were added.
19 July 2004	4.0	Fourth Release. Sections 6,7,8, and the bibliography were updated.
6 December 2004	4.2	Update of the Fourth Release. Sections 6, 8, Tables 16, 17, and the bibliography were updated.
29 June 2005	5.0	Fifth Release. Cookbook Part was completely rewritten. All other parts were updated.
10 August 2005	5.01	Minor update of the 5.0 version. Section 9.6 was added. Sections 9.4.1, 9.5.1 and 9.8 were updated. Table 43 was updated.
15 November 2005	5.1	Update of the Fifth Release. Sections 7.1.2, 7.2,8.3,9.4.2, 9.10.2, 12.2, 12.9, 11, and the bibliography were updated. Section 12.9.2, and Table 55 were added.
12 January 2007	6.0	Sixth Release. Figure 8, sections 5, 7, 8.1, 8.2.3, 12.6, 11, Appendix D, and the bibliography were updated. Sections 9.3, 9.12 were added.
26 September 2007	7.0	Seventh Release. Sections 7, 8, 11 and Appendix D updated.
7 February 2008	7.0	Seventh Release. Typos correction.
10 March 2010	9.0	Ninth Release for OSA 9.0 and ghost_buster description.
12 July 2010	9.1	Update in section "Timing Analysis without the Deconvolution" (ii_pif parameters).
19 July 2010	9.2	Update FOV.
12 September 2012	10.0	Tenth Release for OSA 10.0 Fully revised (modifications are in blue).
3 September 2014	10.1	Release for OSA 10.1. Update of ARF instances in Table 3.
04 SEP 2014	Printed	

Contents

List of Figures

List of Tables

Acronyms and Abbreviations

AD	Architectural Design
ADD	Architectural Design Document
A/D	Analog-Digital
AFEE	Analog Front End Electronics
ASIC	Application Specific Integrated Circuits
BGO	Bismuth Germanate
CdTe	Cadmium-Telluride
CsI	Caesium-Iodide
DBB	Detector Bias Box
DFEE	Digital Front End Electronics
DOL	Data Object Locator
DPE	Data Processing Electronics
DS	Data Structure
FCFOV	Fully Coded Field of View
FIFO	First-In, First-Out
FOV	Field of View
FWHM	Full Width at Half Maximum
GPS	Galactic Plane Scan
GTI	Good Time Interval
GUI	Graphical User Interface
HEPI	Hardware Event Processor
HV	High Voltage
IC	Instrument Characteristics
IJD	Integral Julian Day
ISDC	Integral Science Data Center
ISOC	Integral Science Operations Centre
MCE	Module Control Electronics
MDU	Modular Detection Units
OBT	On-Board Time
OG	Observation Group
PCFOV	Partially Coded Field of View
PEB	PICsIT Electronic Box
PIF	Pixel Illuminated Factor
PMT	Photomultiplier Tube
PLM	Payload Module
RMF	Redistribution Matrix Files
ScW	Science Window
SWG	Science Window Group
TBW	To be written
TM	Telemetry

Glossary of Terms

- *ISDC system:* the complete ground software system devoted to the processing of the *INTEGRAL* data and running at the ISDC. It includes contributions from the ISDC and from the *INTEGRAL* instrument teams.
- *Science Window (ScW):* For the operations, ISDC defines atomic bits of *INTEGRAL* operations as either a pointing or a slew, and calls them ScWs. A set of data produced during a ScW is a basic piece of *INTEGRAL* data in the ISDC system.
- *Observation:* Any group of ScW used in the data analysis. The *observation* defined from ISOC in relation with the proposal is only one example of possible *ISDC observations.* Other combinations of Science Windows, *i.e.,* of observations, are used for example for the Quick-Look Analysis, or for Off-Line Scientific Analysis.
- *Pointing:* Period during which the spacecraft axis pointing direction remains stable. Because of the *INTEGRAL* dithering strategy, the nominal pointing duration is of order of 20 minutes.
- *Slew:* Period during which the spacecraft is manoeuvered from one stable position to another, i.e., from one pointing to another.

1 Introduction

The 'IBIS Analysis User Manual', *i.e.,* this document, was edited to help you with the IBIS specific part of the *INTEGRAL* Data Anaysis.

A more general overview on the *INTEGRAL* Data Analysis can be found in the 'Introduction to the *INTEGRAL* Data Analysis' [1]. For the ISGRI and PICsIT analysis scientific validation reports see [3] and [4].

The 'IBIS Analysis User Manual' is divided into two major parts:

- Description of the Instrument
  This part, based to some extent on the IBIS Observer's Manual of the ISOC AO documentation [2], introduces the *INTEGRAL* on-board Imager (IBIS).
- Description of the Data Analysis
  This part starts with an overview describing the different steps of the analysis. Then, in the Cookbook Section, several examples of analysis and their results and the description of the parameters are given.

  Finally, the used algorithms are described. A list of the known limitations of the current release is also provided.

In the Appendix of this document you find the description of the Raw and Prepared Data and also the description of the Scientific Products.

Part I

Instrument Definition

2 Scientific Performances Summary

IBIS is a gamma-ray telescope observing celestial objects of all classes ranging from the most compact galactic systems to extragalactic objects, with powerful diagnostic capabilities of fine imaging, source identification and spectral sensitivity in both continuum and lines. It is able to localize weak sources at low energy to better than a few arcminutes accuracy, covering the entire energy range from a few tens of keV to several MeV. Table 1 gives an overview of the scientific capabilities of IBIS. The effective area curves are given on the Figure 1.

Table 1: Scientific Parameters of IBIS.

Operating energy range	15 keV – 10 MeV
Energy resolution (FWHM)	7% @ 100 keV 9% @ 1 MeV
Effective Area	ISGRI: 960 cm^2 at 50 keV PICsIT: 870 cm^2 at 300 keV (single events) PICsIT: 275 cm^2 at 1 MeV (multiple events)
Field of view	$8.3° \times 8.0°$(fully coded) $19° \times 19°$(partially coded, 50%)
Angular resolution (FWHM)	$12'$
Point source location accuracy (90% error radius)	$30''$ @100 keV $<5'$ @1 MeV
Continuum sensitivity, photons cm^{-2} s^{-1} keV^{-1} (3σ detection, $\Delta E = E/2$, 10^6 s integration)	3.8×10^{-7} @100 keV $1 - 2{\times}10^{-7}$ @ 1 MeV
Narrow line sensitivity, photons cm^{-2} s^{-1} (3σ, 10^6 s integration)	1.3×10^{-5} @100 keV 4×10^{-5} @ 1 MeV
Absolute timing accuracy (3 σ)	ISGRI: 61 μs PICsIT: 0.976 – 500ms (selected from ground)

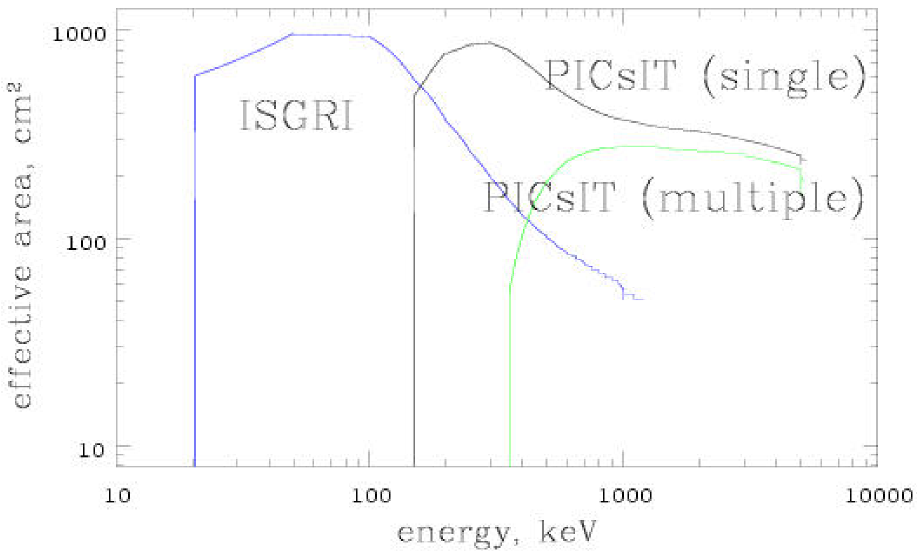


Figure 1: IBIS effective area

3 Instrument Description

3.1 The Overall Design

IBIS is a gamma-ray imager operating in the energy range 20 keV to 10 MeV, with two simultaneously operating detectors covering the full energy range, located behind a Tungsten mask which provides the encoding.

The coded mask is optimized for high angular resolution. As diffraction is negligible at gamma-ray wavelengths, the angular resolution of a coded-mask telescope is limited by the spatial resolution of the detector array. The angular resolution of a coded mask telescope $d\theta$ is defined by the ratio between the mask element size C (11.2 mm) and the mask-to-detection plane distance H (3133 mm).

$$d\theta = \arctan\left(\frac{C}{H}\right) = 12'$$

IBIS is made of a large number of small, fully independent pixels.

The detector features two layers, ISGRI and PICsIT: the first is made of Cadmium-Telluride (CdTe) solid-state detectors and the second of Caesium-Iodide (CsI) scintillator crystals. This configuration ensures a good broad line and continuum sensitivity over the wide spectral range covered by IBIS. The double-layer discrete-element design of IBIS allows the paths of interacting photons to be tracked in 3D if the event involves detection units of both ISGRI and PICsIT. The application of Compton reconstruction algorithms to these types of events (between few hundred keV and few MeV) allows an increase in signal to noise ratio attainable by rejecting those events unlikely to correspond to source photons inside the field of view.

The detector aperture is restricted, in the hard X-ray part of the spectrum, by passive shielding covering the distance between mask and detection plane. An active BGO scintillator VETO system shields the detector bottom as well as the four sides up to the bottom of ISGRI.

Figure 2 shows a cut-away drawing of the various components of IBIS (except the mask and tube). Figure 3 shows the distances between the different parts of the detector assembly. Figure 4 shows the spacecraft & instruments coordinate systems.

3.2 The Subsystems

3.2.1 The Mask

The IBIS Mask Assembly is rectangular with external dimensions of $1180 \times 1142 \times 114$ mm^3, and consists of three main subsystems: the Coded Pattern, the Support Panel and the Peripheral Frame with the necessary interface provisions.

The Coded Pattern is a square array of size $1064 \times 1064 \times 16$ mm^3, made up of 95×95 individual square cells of size 11.2× 11.2 mm^2. The mask chosen for IBIS is based on a cyclic replication of MURA (Modified Uniformly Redundant Array) of order 53. The properties of the MURA patterns are described e.g. in [11] and [12].

Approximately half of the mask cells are opaque to photons in the operational energy range of the IBIS instrument, offering a 70% opacity at 1.5 MeV. The other 50% of cells are open, *i.e.,* with an off-axis transparency of 60% at 20 keV. Figure 5 shows the mask pattern.

The Support Panel includes additional elements to support the code mask pixels, providing the necessary stiffness and strength to overcome the launch environment and the in-orbit operational temperatures. This panel is done from the material known as "nomex". Its transparency should be taken into account in the data analysis, as it absorbs part of the flux.

Figure 6 shows the cross section of the support panel.

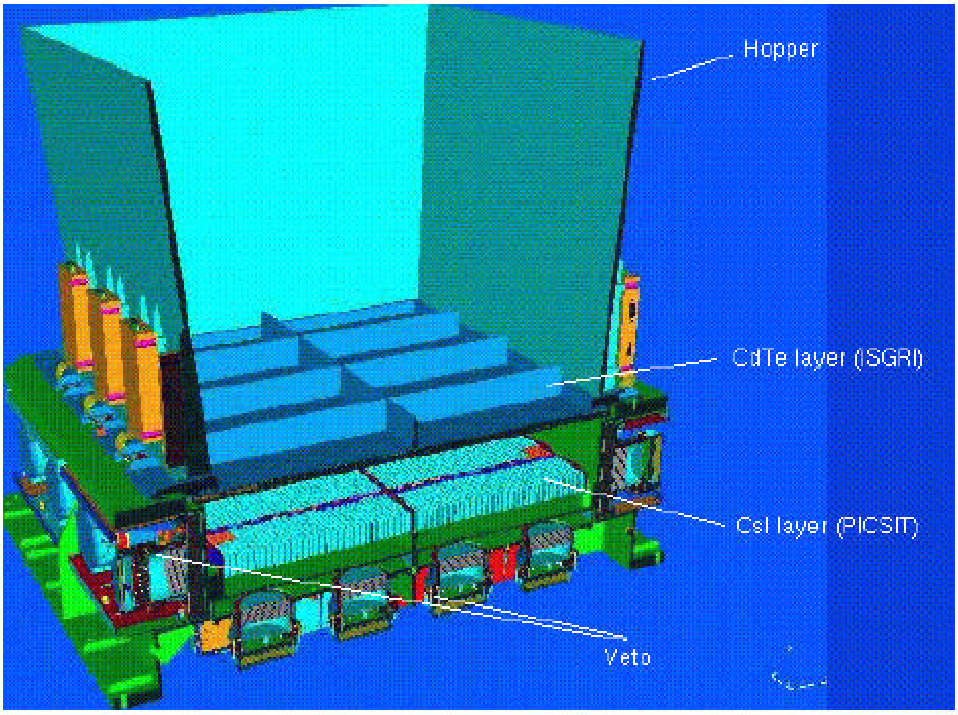


Figure 2: Cutaway drawing of the IBIS detector assembly, together with the lower part of the collimator (Hopper).

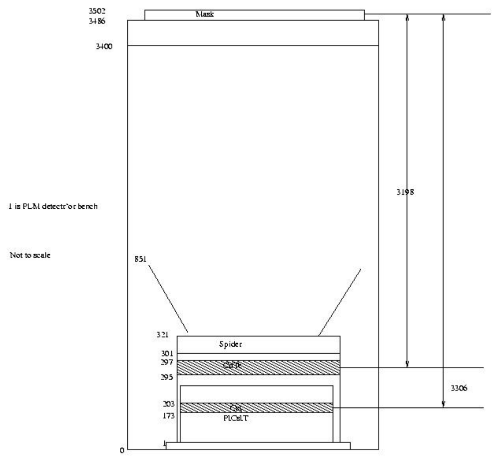


Figure 3: IBIS detector assembly in numbers.

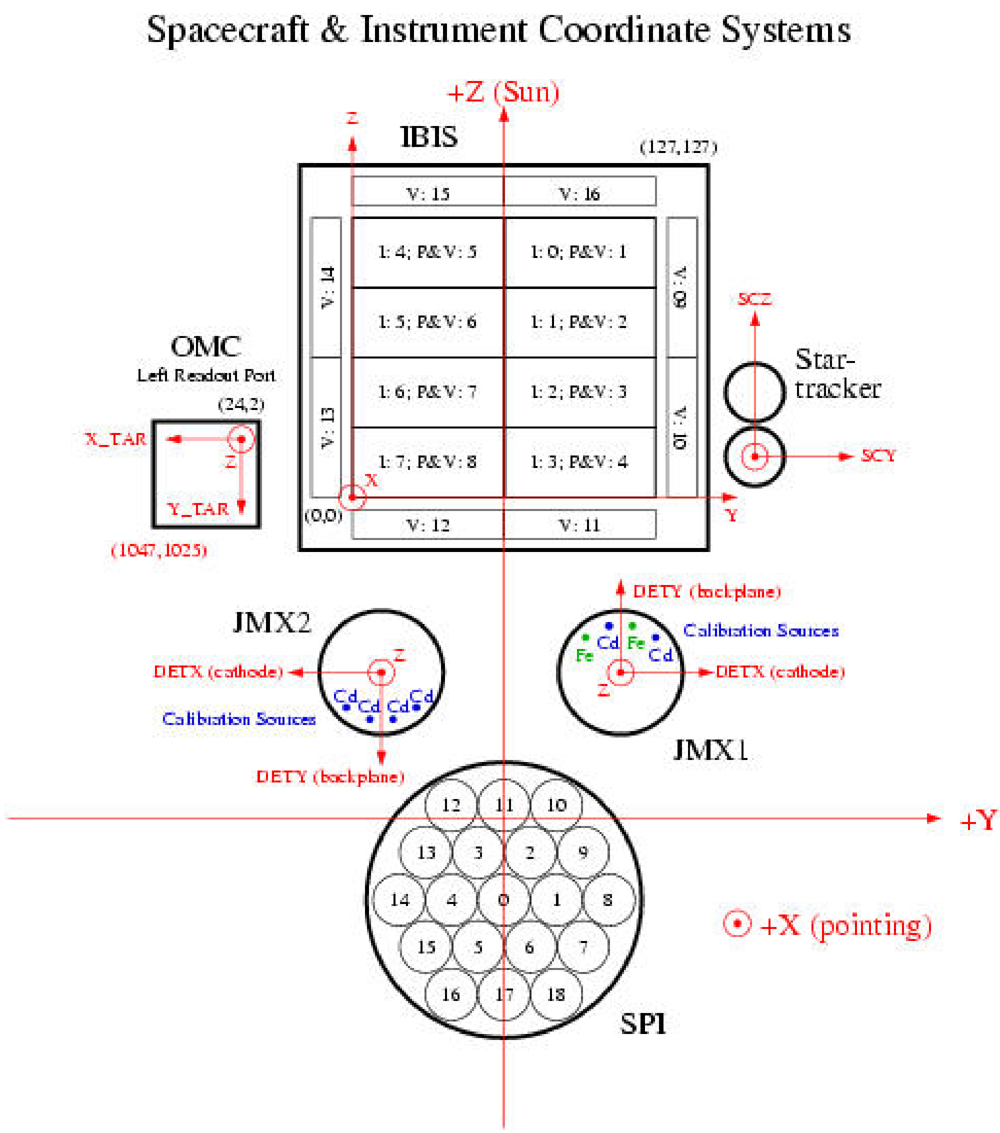


Figure 4: Spacecraft & Instrument Coordinate Systems. Note that the X-axis of the spacecraft is defined by the pointing direction.

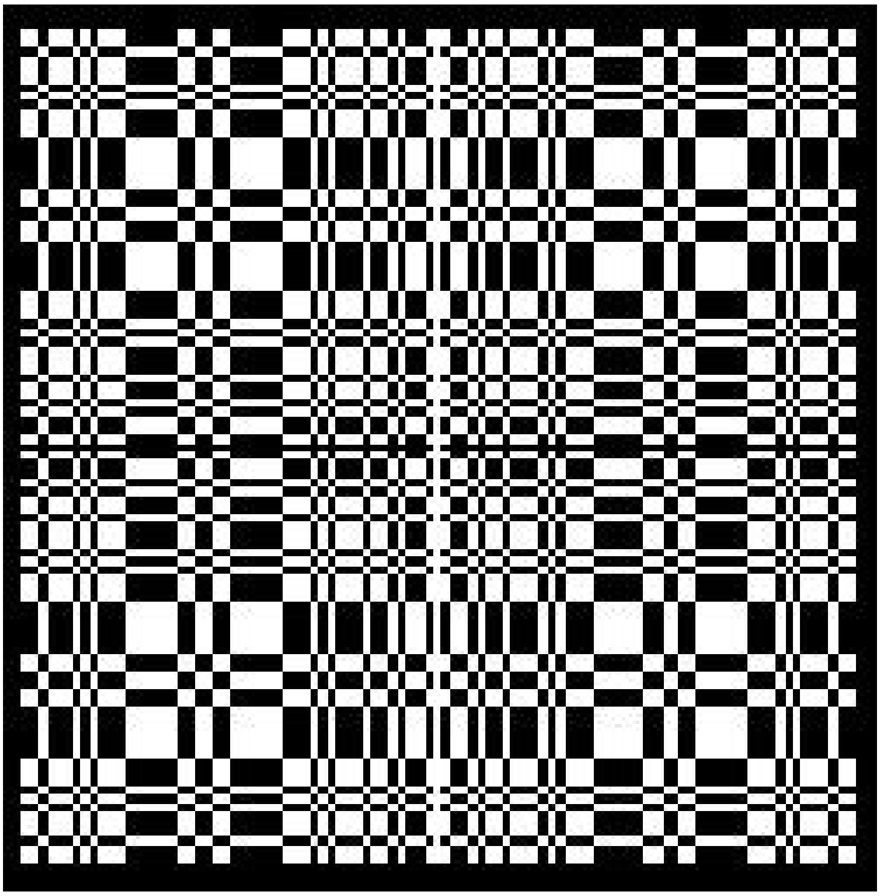


Figure 5: The IBIS coded mask pattern. (white: open elements, back: closed elements)

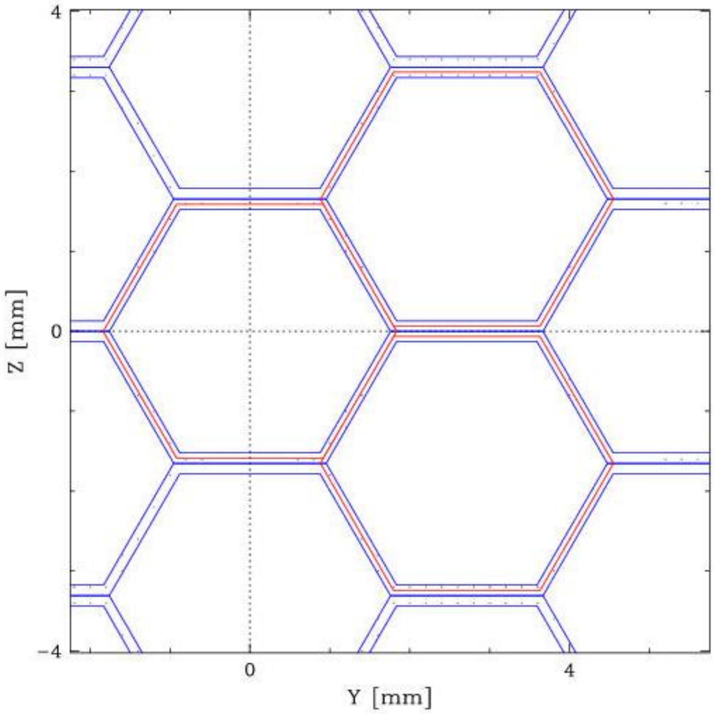


Figure 6: The cross section of the support panel.

The Peripheral Frame reinforces the sandwich panel.

The mechanical interfaces with the INTEGRAL payload module also provide extra Tungsten shielding to the diffuse background through the gap between the mask edges and the payload vertical walls.

3.2.2 The Collimator

In order to maintain the low energy response of IBIS despite the dithering needed for SPI, the collimation baseline consists of a passive lateral shield that limits the solid angle (and therefore the cosmic gamma-ray background) viewed directly by the IBIS detector in the full field of view up to a few hundreds of keV. The tube collimation system is implemented with three different devices:

- *The Hopper*: four inclined walls starting from the detector unit with a direct interface to the IBIS detector mechanical structure. The hopper is not physically connected to the payload module structure.
- *The Tube*: The Tube is formed by four payload module walls shielded with glued Lead foils.
- *The additional side shielding on the mask.* Four strips of 1 mm thick Tungsten provide shielding from the diffuse background in the gaps between the mask edges and the top of the tube walls.

3.2.3 Detector

The ISGRI CdTe and PICsIT CsI(Tl) detectors are layered with respect to each other, with PICsIT below ISGRI with respect to the coded mask (and hence the astronomical source).

- *Upper Detector Layer: ISGRI*

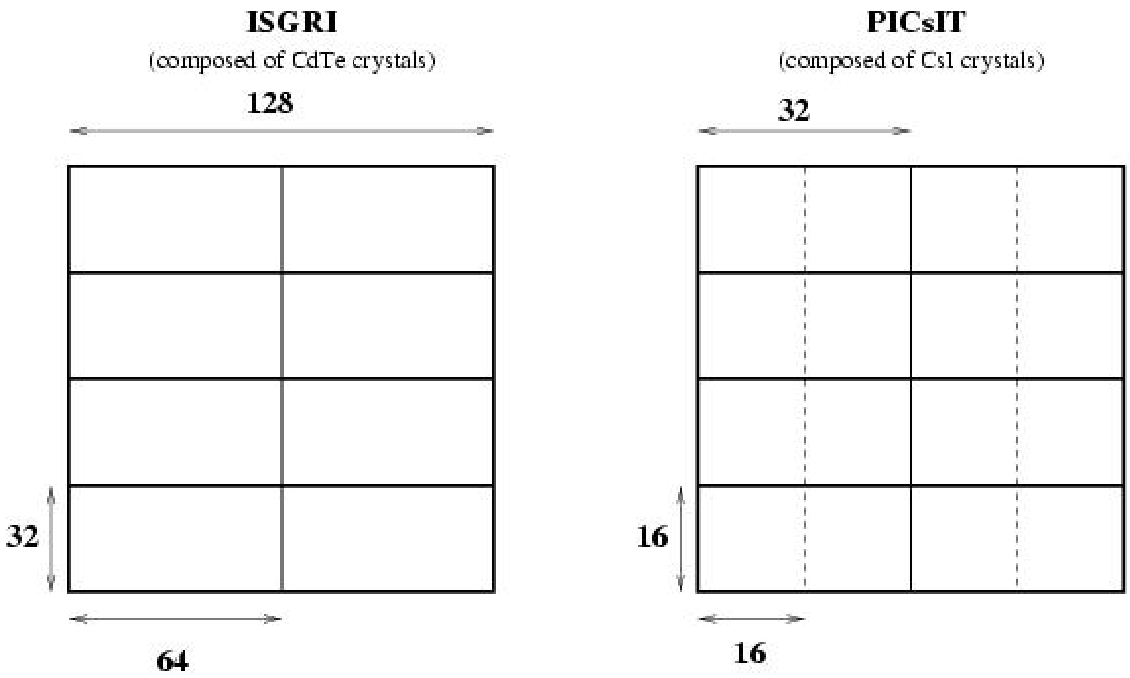


Figure 7: ISGRI and PICsIT division in modules and submodules

Cadmium Telluride (CdTe) is a semiconductor operating at ambient temperature. 0°± 20°C is the optimum range. With their small area, the CdTe detectors are ideally suited to build an image with good spatial resolution.

The CdTe layer is made of 8 identical Modular Detection Units each having 32×64 pixels (see Figure 7). Total sensitive area of the detector is 2621 cm^2.

- *Lower Detector Layer: PICsIT*

  Caesium Iodide is a scintillation crystal. The CsI(Tl) layer is divided into eight rectangular modules of 16×32 detector elements (see Figure 7). In each module there are two independent semi-module each one with its independent Front End Electronics. Total sensitive area of the detector is 2994 cm^2.

- *Noisy Pixels*

  It is possible that with the time some of the pixels of the detector may become out of order and start to produce outputs not triggered by an income photon, *i.e.,* to become "noisy". If the particular pixel countrate is too high relatively to the module countrate, then the on-board electronics switch it off. In ISGRI case the noisy pixels can recover after being switched off for some time and disabled pixels are periodically reset to check their status.

  In PICsIT case, pixels cannot be recovered that easily. PICsIT pixel will remain off once killed. Only if half of the detector (or so) will be off, an attempt will be made to turn pixels on. The current situation is shown on Figure 8. Overall the killed pixels are less than 1%.

3.2.4 On-board Calibration Unit

IBIS contains an on-board collimated radioactive ^{22}Na source. This allows regular calibration of PICsIT at both the 511 keV line (calibration to better than 1% in 4 hours) and 1275 keV (1% in 8 hours). ISGRI can also use the 511 keV line, albeit at lower efficiency. Any energy deposits from untagged photons will have an impact of $<$ 1% on the overall continuum sensitivity between 100 keV and 2 MeV.

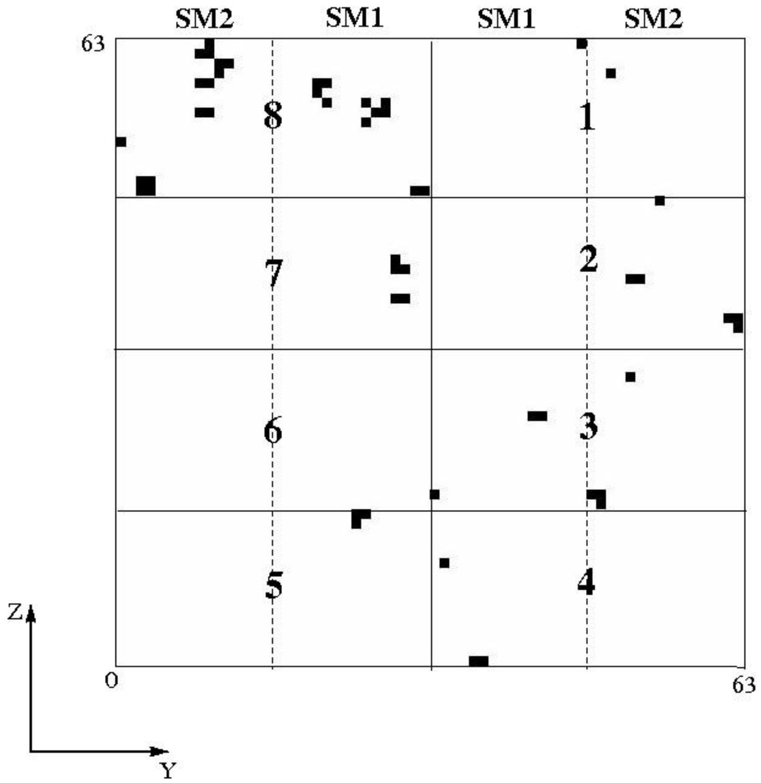


Figure 8: The schematic view of PICsIT layer. Each module number is indicated. The dotted lines represent the division in semimodules whose number is indicated at the top. The black pixels are the killed ones. The (Y,Z) coordinates are the IBIS ones both ranging from 0 to 63. X-axis is directed toward the source located above the page. The Z-axis is pointing positively to the sun.

3.2.5 Veto Shield

The Veto shield is crucial to the operation of IBIS. IBIS uses anticoincidence logic to accept or reject detected events as real photons in the field of view, or background particles or photons propagating through, or induced in, the spacecraft.

The sides, up to the ISGRI bottom level, and rear of the stack of detector planes are surrounded by an active Bismuth Germanate (BGO) veto shield. Like the detector array, the Veto shield is modular in construction. There are 8 lateral shields, *i.e.,* 2 modules per side, and 8 bottom modules.

The high density and mean Z of BGO ensures that a thickness of 20 mm is sufficient to reduce the detector background due to leakage through the shielding of cosmic diffuse gamma-ray background and gamma-rays produced in the spacecraft, to less than the sum of all other background components.

4 How the Instrument works

4.1 Event Types

The photon entering the telescope can be detected due to its interaction with the absorbing material of the detector. Three major types of interactions play a dominant role: photoelectric absorption, Compton scattering and pair production. In the photoelectric absorption process a photon undergoes an interaction with an absorber atom in which the photon completely disappears. In its place an energetic photoelectron is ejected by the atom, carrying away most of the original photon energy. The Compton scattering takes place between the incident gamma-ray photon and an electron in the absorbing material. The incoming photon is deflected and it transfers a portion of its energy to the electron. The energy transferred to the electron can vary from zero to a large fraction of the initial gamma-ray energy. In the pair production process the gamma-ray photon disappears and is replaced by an electron-positron pair. The positron will annihilate in the absorbing medium and two annihilation photons are normally produced as secondary products of the interaction. Depending on the size of the detector and on the energy of the incoming photon, a photon scattered in a Compton interaction can escape the detector, or undergo a second interaction. The pairs of 511 keV photons, produced by the annihilation of the positrons resulting from pair creation, can also produce other interactions or escape the detector.

Both ISGRI and PICsIT record the coordinates of each event registered in the corresponding layer, to build up an image. The anticoincidence VETO is used to reject background events.

The coded mask produces a shadowgram. Photons from the source and the background are distributed across the entire field of view, but cross-correlation techniques allow the full image to be reconstituted for the fully coded field of view ($9° \times 9°$) at each pointing. For the partially coded field of view (out to $29° \times 29°$), special cleaning techniques must be applied to the data to properly reconstruct the image. The actual sky coverage in an observation of course depends on the dither pattern.

The on-board electronics classify registered events according to the activated layer and the number of events detected by a submodule practically simultaneously. Events detected by different submodules are treated as independent ones. There are five main events type:

- *ISGRI single event*

  Photon is stopped in a single pixel of the ISGRI layer, generating an electric pulse.

  In principle, the amplitude of the pulse yields the energy of the incident photon. However, above 50 keV the energy is a function of not just the pulse height but also the pulse rise time, so both are used to determine the energy of the incident photon. In addition the resulting line profile (energy resolution) is no longer Gaussian, but more similar to a Lorentzian. The energy resolution depends on the operating temperature, and also on the bias voltage; the bias voltage has to be optimized as a trade-off between high resolution but more noise (high voltage) or lower noise but lower resolution (low voltage).

  All cases of multiple ISGRI detection units excitation (in one module) are rejected. In case of the excitation of the detection units in different modules, such events are treated as independent single events.

- *PICsIT single event*

  Photon passes through ISGRI and is stopped in a single pixel of the PICsIT layer, generating one scintillation flash.

  The energy of the incident photon is derived, in each crystal bar, from the intensity of the flash recorded in the photodiode. The energy resolution of PICsIT is a function of the signal-to-noise of the event, which in turn is governed by factors operating conditions and PIN capacity.

- *PICsIT multiple event*

  Several PICsIT detection units in one submodule were excited during one event, generating several scintillation flashes. The energy of the primary photon is determined from the sum of the energies of all detected events. The position of the incoming photon is attributed to the position of the most energetic event.

- *Compton single event* Photons arriving in either ISGRI or PICsIT produce secondary photon via Compton scattering, detected in another layer. The position of the incoming photon is attributed to the position of the most energetic event, and the energy is determined as the sum of the detected events energies.
- *Compton multiple event*

  One ISGRI detection unit and several PICsIT detection units in one submodule were excited. As in previous cases the position of the incoming photon is attributed to the position of the most energetic event, and the energy is determined as the sum of the detected events energies.

In Fig.9 the efficiencies of the various detection techniques is shown.

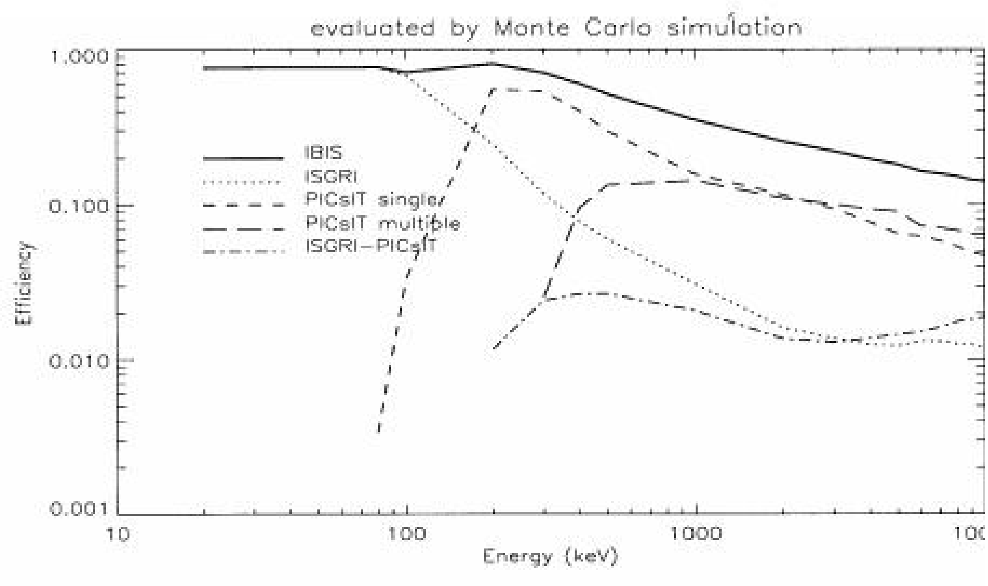


Figure 9: IBIS sensitivities for the various detection techniques.

4.2 IBIS observing modes

IBIS has several observing modes, for engineering and calibration purposes. However, for scientific use there is only one operating mode, Science Mode.

In Science Mode, ISGRI registers and transmits events on a photon-by-photon basis, *i.e.,* every event is tagged with (X,Y) position on the detector plane, event energy (from the pulse height and rise time) and event time.

PICsIT in principle can also operate in photon-by-photon mode. However, with the higher background compared to ISGRI, there would be unacceptable data loss. Therefore, the standard mode for PICsIT is histogram. Images and spectra (full spatial resolution, 256 energy channels) are accumulated for about 30 minutes before transmission to ground. There is no time-tagging internal to the histogram, *i.e.,* spectral imaging has time resolution of 30 minutes.

In addition, coarse spectra, without imaging information, are accumulated by PICsIT and transmitted with far higher time resolution, but without imaging information. Thus their usefulness is limited to observations of very strong sources where the source countrate dominates the background. The time resolution, and the number of energy channels, for this spectral timing data can be commanded from ground. The time resolution can take values between 1 and 500 ms; the current default is 16 ms and eight energy channels.

In Table 2 the properties of all the modes are summarized.

Table 2: Characteristics of the IBIS Telemetry Formats

Observing Mode	Detector Image Resolution (pixels)	Timing Resolution	Spectral Resolution (channels)
ISGRI			
photon-by-photon	128×128	61.035μs	2048
PICsIT			
Photon-by-Photon	64×64	64μs	1024
Spectral-Imaging	64×64	$\leq \sim$30 min	256
Spectral-Timing	None	1 – 500ms	2 – 8

Part II

Cookbook

5 Overview

In this Section an overview of the analysis of IBIS data is given.

Each photon detected by IBIS is analyzed with the on-board electronics and tagged with the arrival time, type (ISGRI , PICsIT, Compton[1] etc.), energy, position etc. according to the operation mode (i.e. photon-by-photon, standard, calibration etc.). These data are then sent to ground in telemetry (TM) packets.

During Pre-Processing the TM packet information is deciphered and rewritten into the set of FITS files (RAW data). Then the local on-board time is converted into the common on-board time (OBT) and the House Keeping (HK) parameters into physical units (PRP data).

These steps are done at ISDC and you do not have to redo them. In the Appendix A and B you will respectively find the description of the raw and prepared data and also the description of the instrument characteristic files that are used in the Scientific Analysis.

INTEGRAL data is organized into the so-called Science Windows (see Introduction to the *INTEGRAL* Data Analysis [1] for more explanations). During the scientific analysis, all the Science Windows belonging to the same observation are grouped together to form the "Observation Group".

Figure 10 shows in details the different steps performed by the scientific analysis script, *ibis_science_analysis*. This high level script consists of four smaller ones: *ibis_scw1_analysis*, *ibis_obs1_analysis*, *ibis_scw2_analysis* and *ibis_obs2_analysis*. *ibis_scw1_analysis* and *ibis_scw2_analysis* work on a Science Window basis while *ibis_obs1_analysis* and *ibis_obs2_analysis* work on the Observation Group basis. Each subscript performs the tasks shown in Figure 10, explained in more details in the text below.

[1] For the time being, Compton analysis is not available

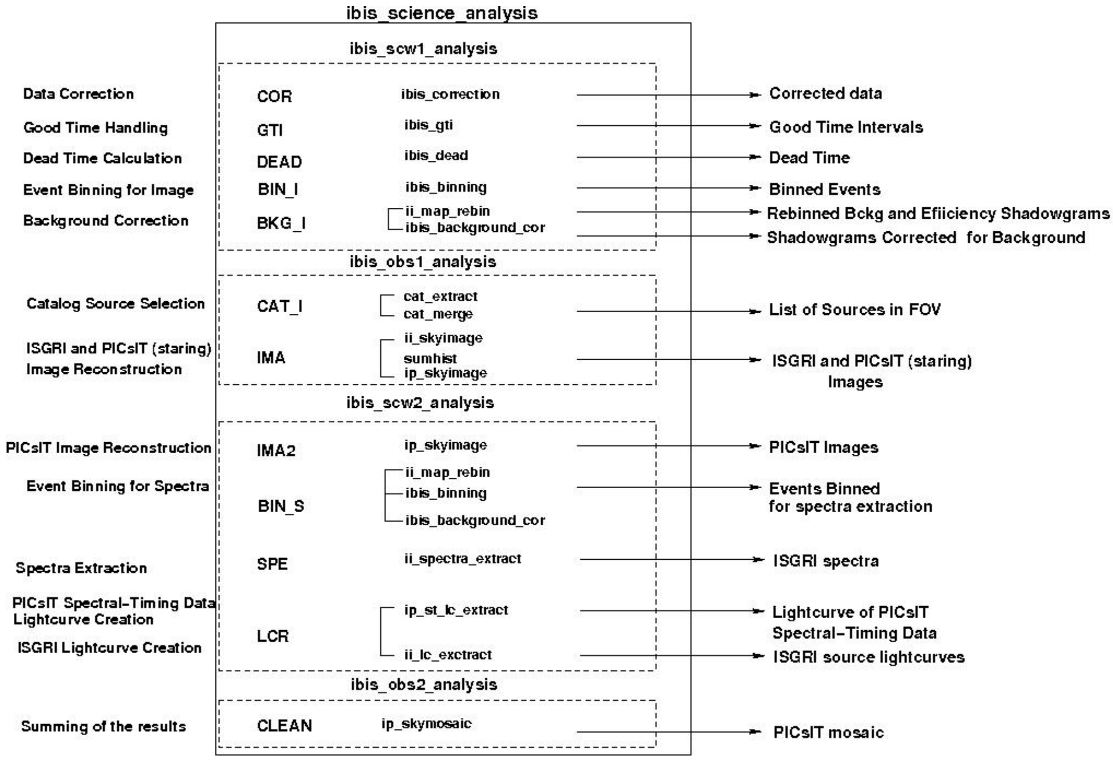


Figure 10: Science Analysis Overview

- The first script *ibis_scw1_analysis* performs the following tasks:

  COR – Data Correction

  Tags noisy pixels, corrects energy of the photons for rise time and temporal variations of the gain, transforms channels to energy. See Section 12.1 for more details.

  GTI – Good Time Handling

  Generates, selects, and merges Good Time Intervals (GTI) to produce a unique GTI that is then used by the software to select good events. See Section 12.2 for more details.

  DEAD – Dead Time Calculation

  Calculates the total dead time during which the incoming photons may be lost due to the processing of the previous events. Also veto strobe signals generated by BGO (Bismuth Germanate) shield, calibration source and Compton events are taken into account. See Section 12.3 for more details.

  BIN_I – Event Binning for Imaging

  Sorts data into energy bins. For each energy range, the intensity shadowgram and a corresponding efficiency map are created. See Section 12.4 for more details.

  BKG_I – Background Correction

  Creates rebinned maps for background and absorption of support mask (see Section 3.2.1) corrections. Corrects for efficiency and subtracts background. See Section 12.6 for more details.

  After these steps the high-level analysis is performed.

- The second script *ibis_obs1_analysis* takes the whole Observation Group previously created as input and performs the following tasks:

  CAT_I - Catalog Source Selection for Imaging

Selects from the given catalog a list of sources in the Field of View matching the criteria defined by script parameters, and creates an output list with location and expected flux values of the selected sources. See Section 12.7 for more details.

IMA - ISGRI and PICsIT (staring) Image Reconstruction

In the case of ISGRI, shadowgrams are deconvolved, source search is performed in the single images as well as in the mosaic (combination of different images) and a list of detected sources is created.

If *INTEGRAL* was stable during the whole period of interest, then, at your request, all PICsIT shadowgrams are combined into one and then are deconvolved into a single image. See Section 12.8 for more details.

- The third script *ibis_scw2_analysis* again works Science Window by Science Window and performs the following tasks:

  IMA2 - PICsIT Image Reconstruction

  PICsIT shadowgram deconvolution is done at this step, creating a separate image for each science window. See Section 12.8.4 for more details. Nothing is done at this step for ISGRI.

  BIN_S – Event Binning for Spectra

  Creates rebinned maps for background and absorption of support mask (see Section 3.2.1) corrections. Sorts data into energy bins. For each energy range the shadowgram and a corresponding efficiency shadowgram is created. See Sections 12.4, 12.6 for more details.

  SPE - ISGRI spectra extraction

  For each source of interest,one PIF[2] is produced. ISGRI spectral extraction is done for all catalog sources with the use of these PIFs. See Section 12.9 for more details.

  LCR - PICsIT Detector Light Curve Creation and ISGRI source lightcurve extraction

  At this step, PICsIT Detector light curves are built from the spectral timing data. For all sources from the input ISGRI catalog light curves are extracted. See Section 12.10 for more details.

- The fourth script *ibis_obs2_analysis* again works in the whole Observation Group previously created, and performs a single task for PICsIT data:

  CLEAN – Last step

  At this step PICsIT mosaic is created. See Section 12.11 for more details.

Revision 3 (rev_3) of the INTEGRAL Archive is completed and is now the default data format. The older Revision 2 (rev_2) archive is no more populated since revolution 1010 (2011-01-20). Rev_3 data have been regenerated starting from INTEGRAL telemetry, including a much improved time correlation and the improved data selection and calibrations used in OSA9. The data correction step (COR), as well as the instrumental GTI and deadtime handling (DEAD) steps have already been performed using a previous version of OSA at the science window level (as was done also for rev_2 data). However, to fully exploit OSA10 improvements, it is mandatory to rerun the analysis from COR step included, as new calibration files are available.

The resulting OSA count rates are corrected for the off-axis transparency of the mask supporting structure.

[2] The Pixel Illumination Factor (PIF) is a number between 0 and 1, which expresses the theoretical degree of illumination of each detector pixel for a given source in the sky.

6 Getting started

This chapter describes how to set up the the environment and the analysis data and how to analyse data from the two instruments that are part of IBIS: ISGRI and PICsIT. These two instruments are quite different in energy range (ISGRI starts from 15 keV and PICsIT starts from 200 keV) and sensitivity, and are optimised for different targets. This is why we have decided to guide you through the analysis of the crowded Galactic Centre around 4U 1700-377 for ISGRI and of the bright Crab for PICsIT.

Here we assume that you have already successfully installed the ISDC Off-line Scientific Analysis (OSA) Software (the directory in which OSA is installed is later on referred to as the ISDC_ENV directory). If not, then look at the "*INTEGRAL* Off-line Scientific Analysis Installation Guide" [5], for detailed help.

6.1 Setting up the analysis data

In order to set up a proper environment, you first have to create an analysis directory (e.g `ibis_data_rep`) and "cd" into it:

```
mkdir ibis_data_rep
cd ibis_data_rep
```

This working directory `ibis_data_rep` will be referred to as the "REP_BASE_PROD" directory in the following. All the data required in your analysis should then be available from this "top" directory, and they should be organized as follows

- **scw/** : data produced by the instruments (e.g., event tables) cut and stored by ScWs;
- **aux/** : auxiliary data provided by the ground segment (e.g., time correlations);
- **cat/** : ISDC reference catalogue (OSA_CAT package);
- **ic/** : Instrument Characteristics (IC), such as calibration data and instrument responses (OSA_IC package);
- **idx/** : set of indices used by the software to select appropriate IC data (OSA_IC package).

The `cat/, ic/` and `idx/` directories are part of the OSA software distribution and should be installed following the "*INTEGRAL* Off-line Scientific Analysis Installation Guide" [5]. The actual data along with the auxiliary files (`scw/` and `aux/`) are sent to the Principal Investigators of the observation. Alternatively, the public data can be downloaded from the archive (see Section 6.1.1). In case the data are already available on your system you can either copy these data to the relevant working directory, or better, create soft links as shown below. Alternatively, if you do not have any of the above data on your local system, or if you do not have a local archive with the `scw/` and the `aux/` branch available, follow the instructions in the next section to download data from the ISDC WWW site.

```
ln -s directory_of_ic_files_installation__/ic ic
ln -s directory_of_ic_files_installation__/idx idx
ln -s directory_of_cat_installation__/cat cat
ln -s directory_of_local_archive__/scw scw
ln -s directory_of_local_archive__/aux aux
```

Then, just create a file "isgri_gc.lst" containing the 5 lines:

```
scw/0051/005100410010.001/swg.fits
scw/0051/005100420010.001/swg.fits
```

```
scw/0051/005100430010.001/swg.fits
scw/0051/005100440010.001/swg.fits
scw/0051/005100450010.001/swg.fits
```

and a file "picsit.lst" containing:

```
scw/0039/003900020020.001/swg.fits
scw/0039/003900020030.001/swg.fits
scw/0039/003900020040.001/swg.fits
scw/0039/003900020050.001/swg.fits
scw/0039/003900020060.001/swg.fits
```

The created files contain the list of ScWs you want to analyze[3].

These file names 'isgri_gc.lst' and 'picsit.lst' will be used later as an argument for the og_create program (see Sections 7, 10).

6.1.1 Downloading data from the archive

- To retrieve the required ISGRI analysis data from the archive, go to the following URL:

  http://www.isdc.unige.ch/integral/archive

  You will reach the W3Browse web page which will allow you to build a list of Science Windows (SCWs) that you will analyse with OSA.

  - Type the name of the object (4U 1700-377) in the '**Object Name Or Coordinates**' field.
  - Do not forget to change '**Search Radius**' if you are interested in science windows where your source is in the partially coded field of view. Set it to e.g. 10 degrees instead of the default 5 degrees = 300 arcmin.
  - Click on '**More Options**' button at the top or at the bottom of the web page.
  - Deselect the '**All**' checkbox at the top of the Catalog table, and select the '**ScW - Science Window Data**' one.
  - Press the '**Specify Additional Parameters**' button at the bottom of the web page.
  - Introduce values in the fields of interest. For instance:
    * Sort output by '**scw_id**', for that check the '**Sort**' column.
    * Put '`pointing`' in the field '**scw_type**' (to specify that only pointings should be returned and not slews)
    * Put '>=`2003-03-15T23:00:00`' in the field '**start_date**' and put '<= `2003-03-16T02:30:00`' in the field '**end_date**'.
    * Put `public` in the field '**ps**' (to specify that only public ScWs should be returned).
    * Put >`100` in the field '**good_isgri**' (to select Science Windows with good ISGRI time higher than 100 seconds)
    * Press the '**Start Search**' button at the bottom of the web page. In our case, a table with 5 ScWs will be displayed.
    * Select the ScWs of interest. To follow the example in the Cookbook click on '**All**' (for all ScWs).
    * Press the '**Save ScW list** for the creation of Observation Groups' button at the bottom of that table and save the file with the name '`isgri_gc.lst`'.

  This file 'isgri_gc.lst' will be used later as input for the *og_create* program (see Section 7). From this file you need the 5 lines below either in the format as given above or simple as:

[3] It is no more needed to add the extension '`[1]`' but the name of the FITS file ('`swg.fits`') is needed unless you use the very basic format without the directory path '`scw/0039/`' and without the final '/' as shown below [1].

```
005100410010.001
005100420010.001
005100430010.001
005100440010.001
005100450010.001
```

You should then download them pressing the '**Request** data products for selected rows' button. In the 'Public Data Distribution Form', provide your e-mail address and press the '**Submit Request**' button. You will get per e-mail the required script to get your data and the instructions for the settings of the IC files and the reference catalogue. Just follow these instructions.

- To retrieve the required PICsIT analysis data from the archive proceed in the same manner with the following parameters:
  - '**Object Name Or Coordinates**' : `Crab`
  - '**Search Radius**' : use default value
  - '**scw_id**' : `0039%` (To select only science windows starting with 0039)
  - '**start_date**' : >=`2003-02-07T06:44:19` and '**end_date**': <=`2003-02-07T12:44:05` and save your results in a file called '`picsit.lst`' which should contain:

    ```
    003900020020.001
    003900020030.001
    003900020040.001
    003900020050.001
    003900020060.001
    ```

6.2 Setting the environment

Before you run any OSA software, you must also set your environment correctly.

The commands below apply to the `csh` family of shells (i.e `csh` and `tcsh`) and should be adapted for other families of shells[4].

In all cases, you have to set the `REP_BASE_PROD` variable to the location where you perform your analysis (e.g the directory `ibis_data_rep`). Thus, type:

```
setenv REP_BASE_PROD  $PWD
```

Then, if not already set by default by your system administrator, you should set some environment variables:

```
setenv ISDC_ENV directory_of_OSA_sw_installation
setenv ISDC_REF_CAT "$REP_BASE_PROD/cat/hec/gnrl_refr_cat_0033.fits"
source $ISDC_ENV/bin/isdc_init_env.csh
```

Note that it is important to ensure that the version of the catalog is at least '0031' (`gnrl_refr_cat_0031.fits`) because it includes new flags that are used by OSA. The latter command executes the OSA set-up script (`isdc_init_env.csh`) which initialises further environment variables relative to `ISDC_ENV`. Ignore all warnings mentioning `ROOTSYS`.

Besides these mandatory settings, the optional environment variable `COMMONLOGFILE` can also be useful. By default, the software logs messages to the screen (`STDOUT`). To have these messages in a file (i.e `common_log.txt`), and make the output chattier[5] use the command:

[4] If the setenv command fails with a message like:'`setenv: command not found`' or '`setenv: not found`', then you are probably using the sh family. In that case, please replace the command '`setenv my_variable my_value`' by the following command sequence: '`my_variable=my_value ; export my_variable`'. In the same manner, replace the command '`source a_given_script.csh`' by the following command '`. a_given_script.sh`' (notice the leading '.' !).

[5] For instance, the exit status of the program will now appear.

```
setenv COMMONLOGFILE +common_log.txt
```

6.3 Two ways of launching the analysis

6.3.1 Graphical User Interface (GUI)

When you launch the analysis the Graphical User Interface (GUI) is launched, providing an opportunity to set the values of all desired parameters, see Figure 12. On the right side of the panel you see the following buttons:

- `Save as` : button to create a file storing all parameters as they are currently defined in the GUI as a command line script. The file can be executed from the command line to launch the instrument analysis program with the parameters as they were defined in the GUI.
- `Load` : button to read a previously saved file (with '`Save As`') that will be used to update all parameters of the GUI with the values defined in the loaded file.
- `Reset` : button to reset all parameters in the GUI to their default values as they are defined in the parameter file of the instrument analysis program and stored in the `$ISDC_ENV/pfiles/` directory.
- `Run` : button to launch the analysis with the parameters currently defined in the GUI.
- `Quit` : button to quit the program without launching the analysis.
- `Help` : button to open the help file of the main script in a separate window.
- `hidden` : button to access the hidden parameters with values defined by the Instrument Team. Change them with care!

6.3.2 Launching scripts without GUI

Instead of using the GUI, parameters can be specified on the command line typing '`name = value`' after the script name.

If you are running your own scripts that call OSA many times you don't want the GUI to pop up each time. In such a case set `COMMONSCRIPT` variable to '1' with:

```
setenv COMMONSCRIPT 1
```

This is automatically done if you use the file created with the help of the '`Save as`' - button, see above.

To have the GUI back again, unset the variable:

```
unsetenv COMMONSCRIPT
```

6.4 Useful to know!

- How do I get some help with the executables?
  All the available help files are stored under `$ISDC_ENV/help`. To visualize a help file interactively type `tool_name --h` once your environment is set (i.e. the command `which tool_name` properly returns the path to it).
- Where are the parameter files and how can I modify them?
  All the available executables for the analysis of *INTEGRAL* data are under `$ISDC_ENV/bin/`. The corresponding parameter files are stored under `$ISDC_ENV/pfiles/*.par`. The first time you launch

a script, the system will copy the specific `tool.par` from `$ISDC_ENV/pfiles/` to a local directory (~`user_name/pfiles/`). The parameter file in the local directory is the one used for the analysis and is the one you can modify. If this parameter file is missing (e.g. you have deleted it), the system will just re-copy it from `$ISDC_ENV/pfiles/` as soon as you launch the script again. When installing a new version of OSA, it is safe to update your local instance of pfiles, because some default parameter values may have changed and using the old ones may badly influence the results of your analysis. You can do this with: `cp -r $ISDC_ENV/pfiles/ ~/`. Removing them instead is not a good solution as they will be missed when running *og_create* (see Sect. 7). The system knows what to copy from where thanks to the `$PFILES` environment variable that is also used in (FTOOLS). Each parameter is characterized with a letter that specifies its type, i.e:

  – '`q`' (query) parameters are always asked to the user
  – '`h`' (hidden) parameters are not asked to the user and the indicated value is used
  – '`l`' (learned) parameters are updated with the user's value during the use of the program

The GUI is a fast and easy way to change the parameters, see section 6.3.1 for details.

- What are groups and indices?
  The ISDC software makes extensive use of groups and indices. While it is not necessary to grasp all the details of these concepts, a basic understanding is certainly quite useful.

  As implied by their names, "groups" make possible the grouping of data that are logically connected. Groups can be seen as a kind of data container, not completely unlike standard directories. At ISDC, we create separate groups for each pointing, in which we store the many different data types produced by INTEGRAL and its instruments. The user then only has to care about one file, the group, many tens of files being silently included. Several pointings (the "Science Window Groups") can be arbitrarily grouped into bigger groups (the "Observation Group") to select data very efficiently according to the user's needs.

  Indices are a special kind of groups, which differ only in the fact that all the the data sets they contain are similar and that the indices know the properties of the data sets they contain. Indices are a kind of poor man's database. For example, an imaging program creates several images of different types (flux map, significance map,...) in different energy bands. These images are stored in an index, in which the image type and energy band information is replicated. ISDC software is then able to select very efficiently the needed images. The user can also make use of the indices; just by looking at the index (for instance using "fv"), the user can identify immediately the content of each image.

- Why do I sometimes have to add a '[1]' after a FITS file name?
  A FITS file can have many extensions and sometimes it is necessary to specify as input to a given parameter not the file name alone (`file.fits`) but the extension too (`file.fits[1]`, or `file.fits[2]`, etc). The file name with a specified data structure (extension) is called a Data Object Locator (DOL). When you modify the parameter file itself (see above) or use the GUI, the extension will be correctly interpreted in the `file.fits[1]` case. On the command line though, the normal CFITSIO and FTOOLS rules apply, i.e. you have to specify it as one of the following:
  `file.fits\[1]`
  `file.fits+1`
  `"file.fits[1]"`.
  Note that the numbering starts with '[0]' but often this extension does not contain real data. If no extension is specified explicitly the first one ('[1]') will thus be used by default.

You will find many more useful information in the ISDC list of Frequently Asked Questions (FAQ) at:
http://www.isdc.unige.ch/integral/support/faq

7 A Walk through ISGRI Analysis

After setting up the OSA environment as described in the previous section, you are ready to analyse the data.

Please do remember that you are dealing with a coded mask instrument not with a focusing telescope and a CCD. It is not possible to deal with one source at a time: each source is background for the others, the whole field of view - and not just the few pixels around your source - matters!

In this Section, we guide you through your first IBIS analysis, but please read also Section 8, where more details on the main parameters are given. You could end up with fake sources that are created by a blind use of parameters! More tips and tricks are given in Section 9 for advanced users.

In the example below we analyze observations of the Galactic Center, using data we have downloaded and installed as it is described in Section 6.1.

Create the Observation Group with the *og_create* program (see its description in the Toolbox section of [1]):

```
cd $REP_BASE_PROD
og_create idxSwg=isgri_gc.lst ogid=isgri_gc  baseDir="./" instrument=IBIS
```

As a result of the *og_create* command, the directory **$REP_BASE_PROD/obs/isgri_gc** is created. In this directory you find all you need for the analysis, its structure is illustrated in Figure 11.

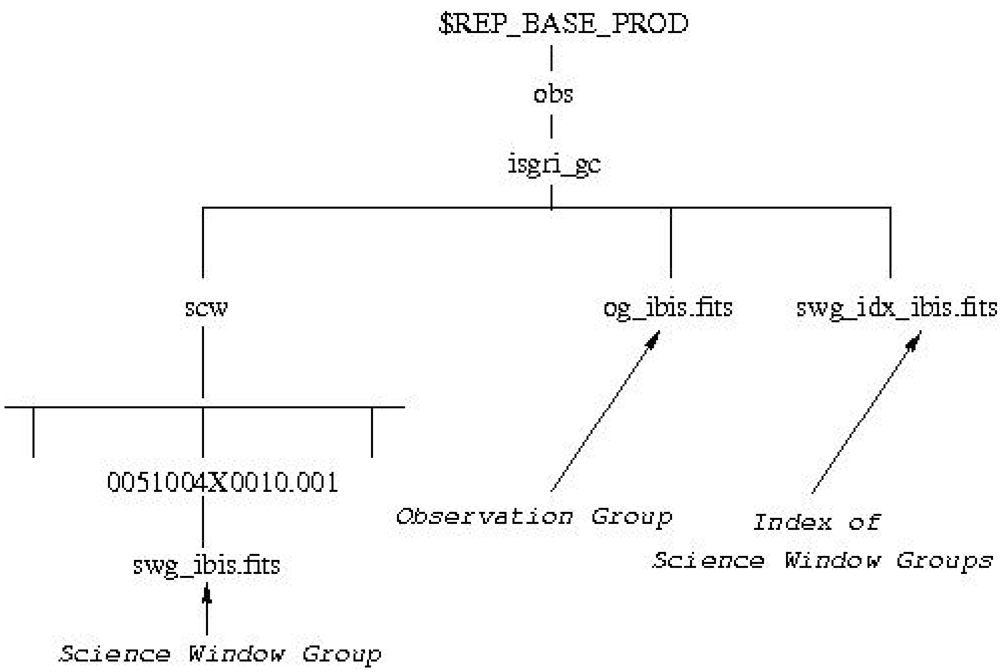


Figure 11: Structure of the directory created with *og_create*

7.1 Image Reconstruction

The first thing to do when you are looking for the first time at your data is to create an image in order to know how the portion of sky you are interested in looks like, whether your source is detected, and what other sources you should take into account to do spectral and lightcurve analysis in a proper way.

To start the analysis, move to the working directory **$REP_BASE_PROD/obs/isgri_gc** and call the *ibis_science_analysis* script:

```
cd obs/isgri_gc
ibis_science_analysis
```

After a few seconds the main page of the IBIS Graphical User Interface (GUI) appears, as shown in Figure 12.

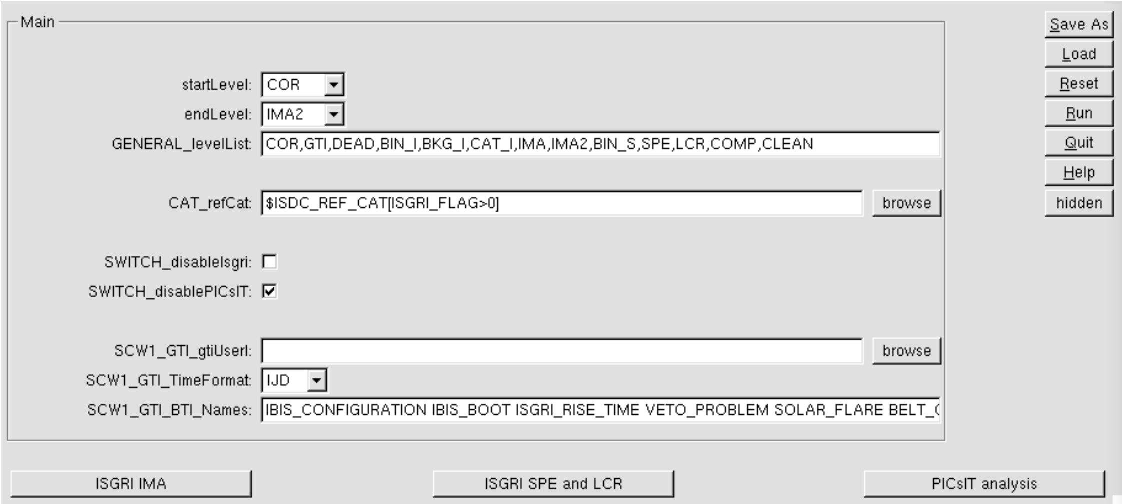


Figure 12: Main page of the IBIS GUI

Keeping all the default values, you will make an analysis starting from the energy correction level (startLevel = COR[6]) until the image reconstruction level (endLevel = IMA2 [7]). The default input catalog (`CAT_refCat= $ISDC_REF_CAT[ISGRI_FLAG>0]`) will be used: it contains the sources that were detected by ISGRI in the public data prior to the release of the catalog. ISGRI data alone will be processed (`SWITCH_disablePICsIT=yes`) through all the available levels shown in GENERAL_levelList within COR and IMA2. No additional, user defined, good time interval will be applied to the data: `SCW1_GTI_gtiUserI` field is empty, which also implies that the next parameter `SCW1_GTI_TimeFormat` is not effective. The last parameter `SCW1_GTI_BTI_Names` specifies types of problems which should lead to exclusion of the science window from the analysis. The detailed description of all types is given in the Section 9.3. The default value is the most conservative and includes all possible categories.

You are ready to set the parameters specific for imaging. Press the **ISGRI_IMA** button at the bottom of the GUI. Another box, shown in Figure 13 appears.

With the default parameters displayed in the GUI, you will create four different images of the sky, corresponding to four energy bands (IBIS_II_ChanNum=4) in the ranges 20–40, 40–60, 60–100 and 100–200 keV. You will let the software look for all catalog sources and up to 50 brightest sources in the field of view (`OBS1_Searchmode=3` and `OBS1_ToSearch=50`), with detection significance higher than 6 for new sources (`OBS1_MinNewSouSnr=6`). Note that the `OBS1_ToSearch` parameter is set to a high value to be safely used to detect all sources in the final mosaic image even in the crowded field around the Galactic Center (cf. Table 19). The position of all the catalog sources will be fitted (`OBS1_SouFit=0`), except for sources with `ISGRI_FLAG==2` (whose position is known with an accuracy better than 3 arcs) where the fixed catalog position will be used.

Resulting images will be cleaned with the available background maps provided by the IBIS team (empty value

[6]Remember that -in order to fully exploit the improvements made in OSA 10- the analysis needs ALWAYS to be performed starting from COR level, as new calibration files have been introduced.

[7]IMA2 level is for PICsIT analysis only, and there is no difference for ISGRI analysis whether endLevel is set to IMA, or IMA2.

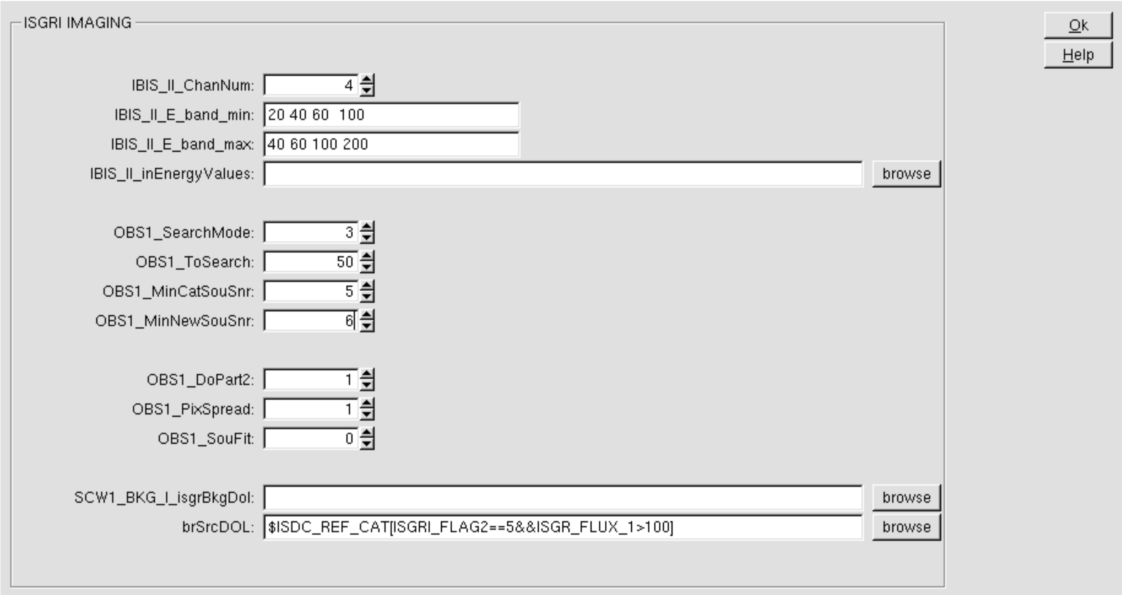


Figure 13: Imaging page of the IBIS GUI

of `SCW1_BKG_I_isgrBkgDol` means usage of the default map for a given Science Window). We choose the background normalization to be calculated from the shadowgrams from which the pixels affected by the photons from all sources previously detected by ISGRI with a catalog flux in the 20–60 keV energy band higher than 100 cts/sec (∼ 600 mCrabs) were removed (`brSrcDOL="$ISDC_REF_CAT[ISGRI_FLAG2==5&&ISGR_FLUX_1>100"]`). In principle, the set of bright sources taken into account for the background subtraction, should be chosen individually for each analysed observation, see more details in Section 8.2.3.

We have found that some regions of the mask have glue deposits in the void. The shadowgrams of strong sources near those glue deposits are not fully correct and create artifacts in the images (mostly ghosts). The *ghost_buster* script — first included in OSA 9 — is called by the analysis script to kill affected pixels from the shadowgrams of the strongest sources in the image. If you include too many sources in this *ghost_buster* algorithm you will also kill too many signal pixels, so only include very bright source and do this if you see artifacts in very deep mosaics.

After the creation of all the individual Science Window (ScW) images the mosaic image will be created (`OBS1_DoPart2=1`). In this mosaic image the photons belonging to a source are spread around the single central peak, resulting in better source location (`OBS1_PixSpread=1`).

Now that you have checked all these parameters, press **Ok**, the Imaging window disappears and you are back to the main GUI page. Press **Run** to launch the analysis script.

7.1.1 Results from the Image Step

When the analysis is finished, you will find that new files have appeared in your working directory. In Figure 14 an overview of the files related to the image reconstruction is given. The full description of all files produced at different levels is given in Appendix C.

As it is shown in Figure 14, there are results for each science window as well as for the overall group.

Science window results include:

- **Images:** `scw/0051004X0010.001/isgri_sky_ima.fits` (where X=1..5)

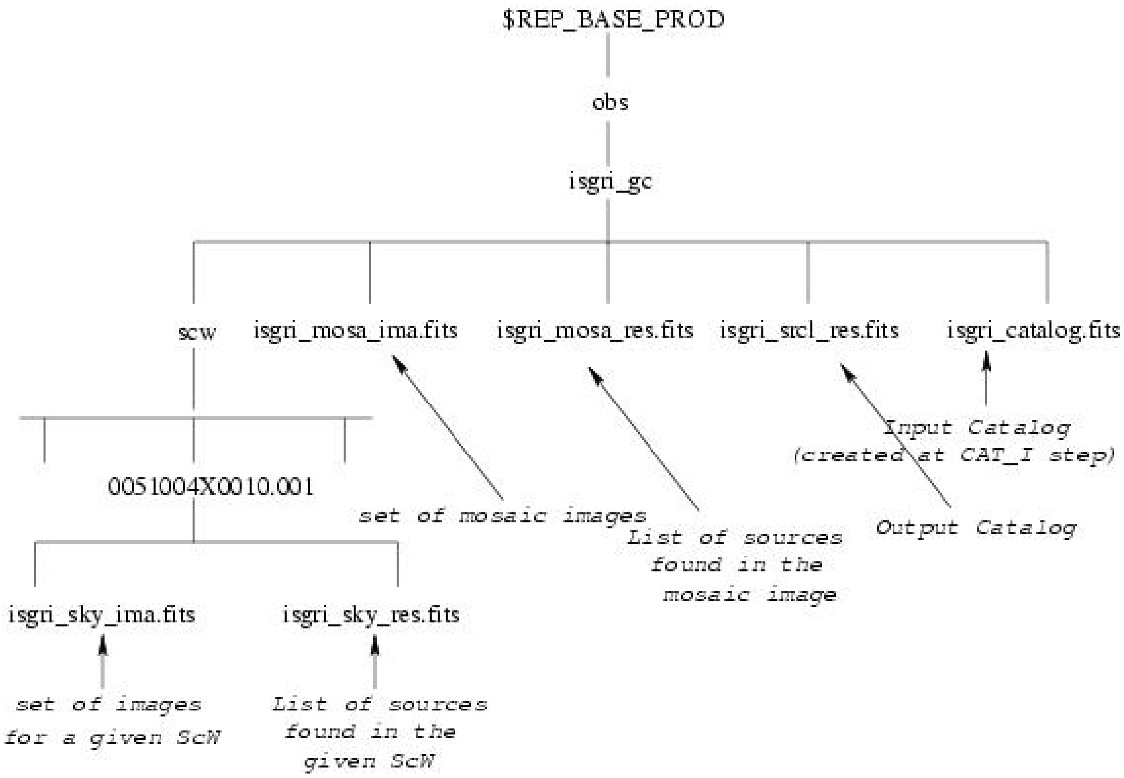


Figure 14: Overview of the IMA level products

for each energy range five images are produced: INTENSITY, VARIANCE, SIGNIFICANCE[8] and RESIDUAL[9] and EXPOSURE[10]. To know which type of image and which energy band correspond to a given extension, you can either check in the header of the `isgri_sky_ima.fits` file extensions, or check the first extension of the file, an index, summarizing the file content. In Figure 15 the column highlighted in green was added by hand to identify the corresponding extensions of the file. More details on index concept are in [1].

- **Detected source list:** `scw/0051004X0010.001/isgr_sky_res.fits` (where X=1..5)
  list of the sources detected in each energy range with reconstructed RA, DEC, flux, error and significance.

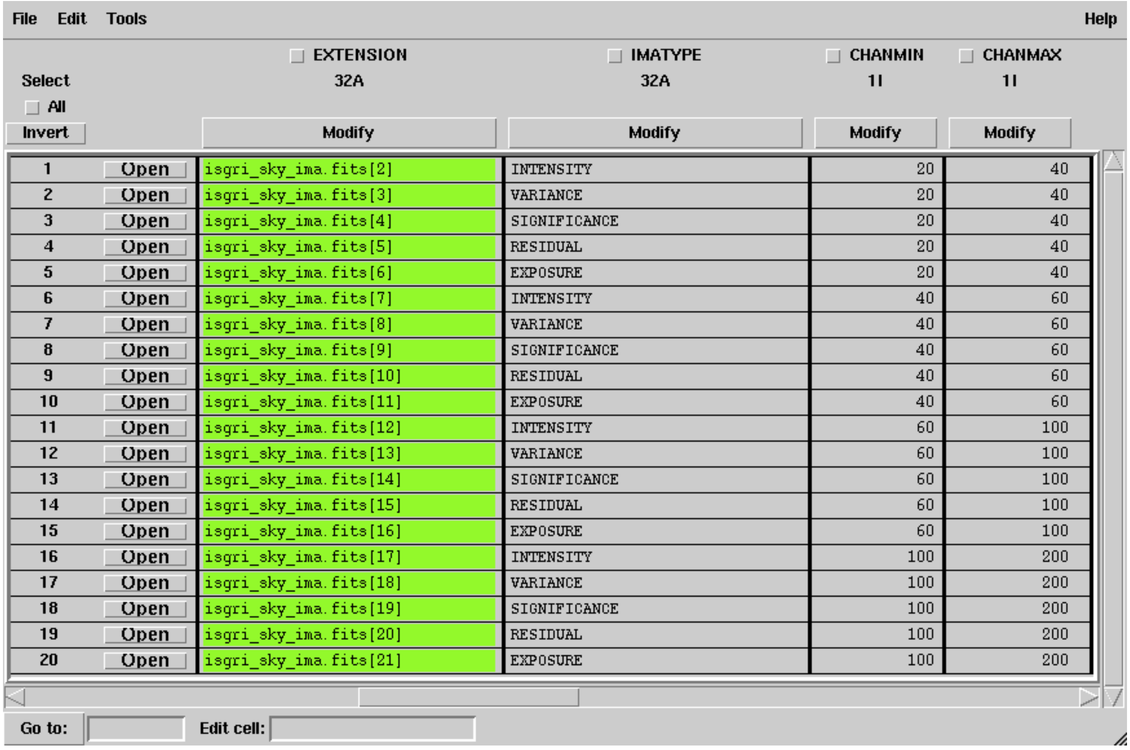
Observation group results include:

- **Image:** `isgri_mosa_ima.fits`
  The structure of this file with the mosaic images is similar to the one of isgri_sky_ima.fits, but instead of residual maps you have the exposure ones.
- **Detected source list:** `isgri_mosa_res.fits`
  list of the sources detected in the mosaic image in each energy range with reconstructed RA, DEC, flux, error and significance.

The single science window and mosaic results are merged in the file `isgri_srcl_res.fits`. This file contains all the sources from the `isgri_catalog.fits` plus all the new sources, with the information on their fitted position, fluxes, and detection level.

You find more details on the structure of the output files in Section C.7.1.

[8] Note that systematic errors are not included in the calculation of the detection significance (DETSIG).
[9] The intensity image from which all photons attributed to the detected sources were removed.
[10] The actual exposure map of the field of view in seconds.



File Edit Tools Help

Select		EXTENSION	IMATYPE	CHANMIN	CHANMAX
All / Invert		32A	32A	1I	1I
1	Open	isgri_sky_ima.fits[2]	INTENSITY	20	40
2	Open	isgri_sky_ima.fits[3]	VARIANCE	20	40
3	Open	isgri_sky_ima.fits[4]	SIGNIFICANCE	20	40
4	Open	isgri_sky_ima.fits[5]	RESIDUAL	20	40
5	Open	isgri_sky_ima.fits[6]	EXPOSURE	20	40
6	Open	isgri_sky_ima.fits[7]	INTENSITY	40	60
7	Open	isgri_sky_ima.fits[8]	VARIANCE	40	60
8	Open	isgri_sky_ima.fits[9]	SIGNIFICANCE	40	60
9	Open	isgri_sky_ima.fits[10]	RESIDUAL	40	60
10	Open	isgri_sky_ima.fits[11]	EXPOSURE	40	60
11	Open	isgri_sky_ima.fits[12]	INTENSITY	60	100
12	Open	isgri_sky_ima.fits[13]	VARIANCE	60	100
13	Open	isgri_sky_ima.fits[14]	SIGNIFICANCE	60	100
14	Open	isgri_sky_ima.fits[15]	RESIDUAL	60	100
15	Open	isgri_sky_ima.fits[16]	EXPOSURE	60	100
16	Open	isgri_sky_ima.fits[17]	INTENSITY	100	200
17	Open	isgri_sky_ima.fits[18]	VARIANCE	100	200
18	Open	isgri_sky_ima.fits[19]	SIGNIFICANCE	100	200
19	Open	isgri_sky_ima.fits[20]	RESIDUAL	100	200
20	Open	isgri_sky_ima.fits[21]	EXPOSURE	100	200

Go to: Edit cell:

Figure 15: A part of the GROUPING extension table of an `isgri_sky_ima.fits` file.

7.1.2 Displaying the Results from the Image Step

It is convenient to look at the images with the help of the *ds9* program. First create a region file from the catalog using the *cat2ds9* program. In the example below we create the region file `found.reg` with all the sources found in the mosaic image, `isgri_mosa_res.fits`, for the first energy band (extension [2]), and a region file `cat.reg` with all sources that were in the input catalog `isgri_catalog.fits`.

```
cd $REP_BASE_PROD/obs/isgri_gc
cat2ds9 isgri_mosa_res.fits\[2] found.reg symbol=box color=green
cat2ds9 isgri_catalog.fits cat.reg symbol=box color=white
```

To see the resulting images:

```
ds9 $REP_BASE_PROD/obs/isgri_gc/isgri_mosa_ima.fits\[2] \
    -region $REP_BASE_PROD/obs/isgri_gc/cat.reg \
    -cmap b -scale sqrt -scale limits 0 60 -zoom 2 \
    $REP_BASE_PROD/obs/isgri_gc/isgri_mosa_ima.fits\[4] \
    -region $REP_BASE_PROD/obs/isgri_gc/found.reg \
    -cmap b -scale sqrt -scale limits 0 60
```

In Figure 16 you see the INTENSITY (left, `$REP_BASE_PROD/obs/isgri_gc/isgri_mosa_ima.fits[2]`) and the SIGNIFICANCE (right, `$REP_BASE_PROD/obs/isgri_gc/isgri_mosa_ima.fits[4]`) mosaic images in the 20–40 keV energy range. In the left image we have shown all catalog sources (white boxes), and in the right one only the detected ones (green boxes). The color scale at the bottom gives the significance values. Although we used here a square-root scaling (`sqrt`) that enhances the structure in the low-values (the background) we now have a very clean mosaic image compared to images obtained with OSA versions prior to OSA 9. The issue of spurious new sources detected by the software at the position of the "ghosts" of true sources is therefore much reduced.

Indeed, in a coded mask instrument with a symmetric mask pattern as in the case of IBIS a true point source will cause secondary lobes, 8 main "ghosts" aligned with the detector edges, at a distance that is a multiple of the mask basic pattern, 10.7 degrees in IBIS/ISGRI case (cf. Figures 5 and 32). The "ghosts" of sources detected in individual ScW images will be removed from these images and will thus not affect the mosaic image. However, if a source is too weak to be automatically detected in a single ScW, its ghosts are not cleaned, they can appear in the mosaic image and even be found by the software as new sources.

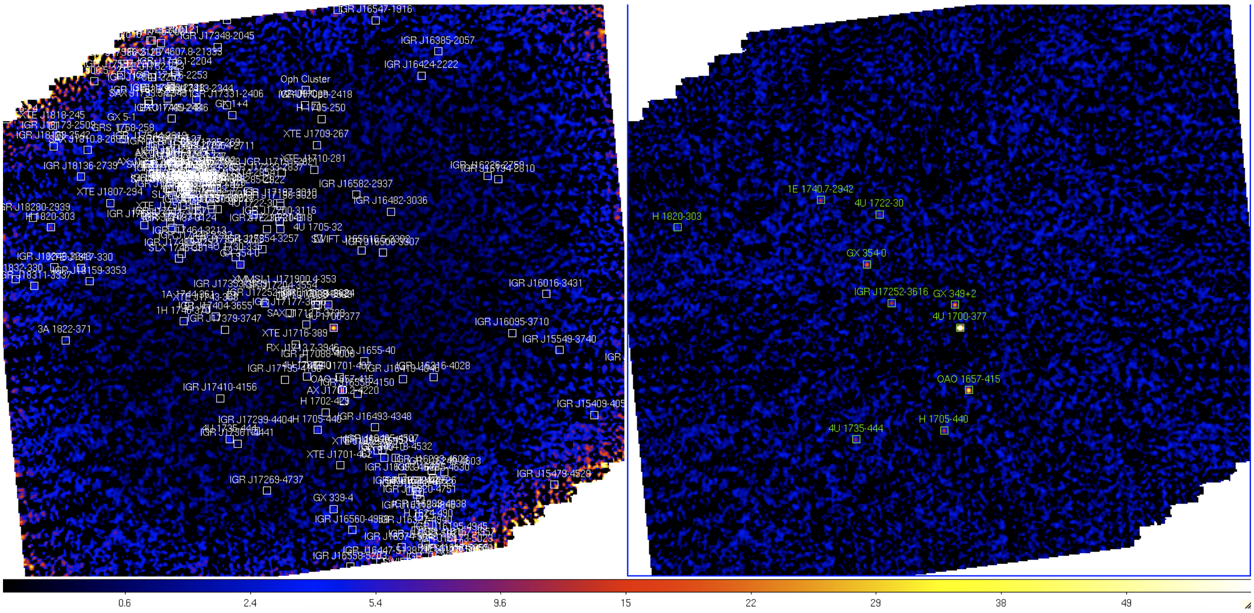


Figure 16: INTENSITY (left) and SIGNIFICANCE (right) mosaic images in the 20–40 keV energy band.

There is an easy way to collect from different ScWs all the information related to a given source and energy band. In the example below we create a file **4U1700-377_scwlc.fits** with all the information on 4U 1700-377 in 20–40 keV energy band. The structure of this file is explained in Appendix C.7, Table 55.

```
src_collect group=og_ibis.fits+1 results=4U1700-377_scwlc.fits \
                    instName=ISGRI select="NAME == '4U 1700-377' && E_MIN==20"
```

In Figure 17 the ScW-per-ScW lightcurve of 4U 1700-377 in the 20–40 keV band is shown. A ligthcurve with a finer time binning can be constructed at the LCR level (see Sect. 7.3). Note that the count rates are already corrected for instrumental effects such as the off-axis transparency of the mask supporting structure.

7.2 Spectral Extraction

It is not possible to extract the spectrum of only the source you are interested in. All sources brighter or compatible with the one you are interested in should be taken into account too. Thus it is strongly recommended, when you deal for the first time with your data, to run the analysis until the IMA2 level, as described in Section 7.1, check the results, and call *ibis_science_analysis* once again to run the spectral extraction part. The description of the algorithm used for spectral extraction is given is Section 12.9.1.

Launch *ibis_science_analysis*, and on the main GUI page change Start Level to **BIN_S**, and End Level to **SPE**. After that press the **ISGRI SPE and LCR** button.

On the screen that appears (see Figure 18), you can specify:

Spectral energy binning:
With the help of the parameter `IBIS_SI_inEnergyValues` you can specify the file (and its extension) describing the desired binning of the response matrix. By default (set with empty line), the latest available file

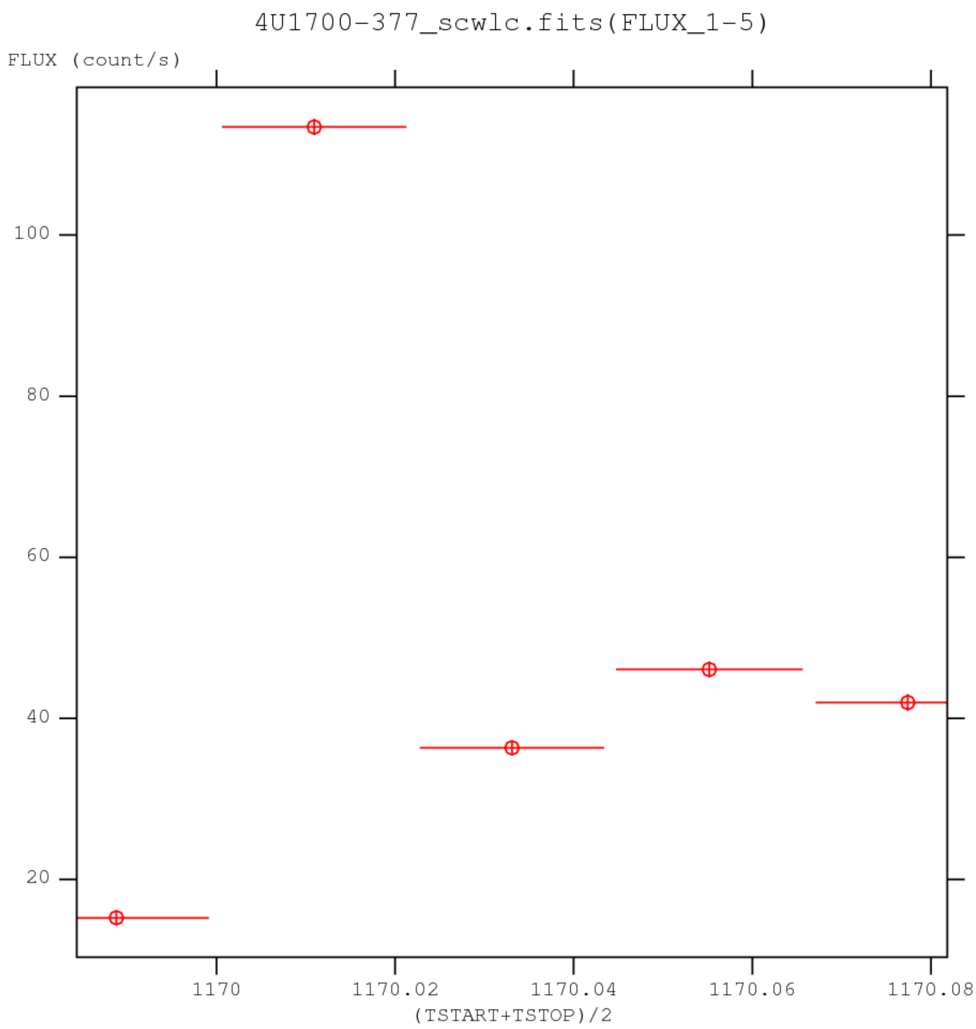


Figure 17: ScW-per-ScW lightcurve of 4U 1700-377 in the 20–40 keV energy band

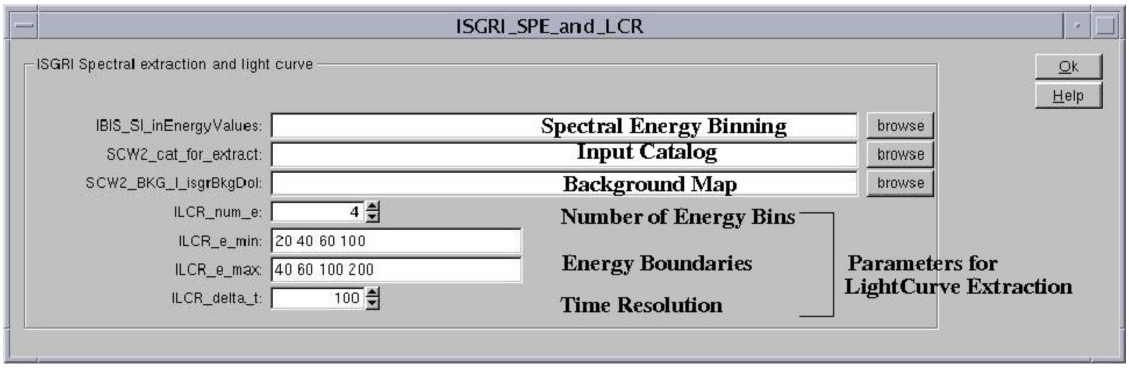


Figure 18: Page of IBIS GUI for Spectral and Ligtcurve extraction

with the rebinned response matrix is used. See sections 8.3 and 9.5 if you want to define your own spectral binning.

Background maps:
With the **SCW2_BKG_I_isgrBkgDol** parameter you can specify the background map to be used in the spectral and timing analysis. It is recommended to use the same background map as in the imaging case. Default empty value corresponds to the default map ($REP_BASE_PROD/ic/ibis/bkg/isgr_back_bkg_0011.fits).

Input catalog for Spectral and Lightcurve extraction:
By default a spectrum will be created for all sources listed in the `isgri_srcl_res.fits` catalog created during the IMA step. The position of the source will be taken from RA_FIN, DEC_FIN columns. Note however that spectral extraction is time consuming, and it is not recommended to include sources that are faint in comparison with the source of interest. Moreover **it is not recommended to have more than 30 sources in the input catalog**, as the software might crash otherwise. Copy `isgri_srcl_res.fits` to another file, say `specat.fits` as shown below, leaving only sources with significance higher than a threshold ($7\,\sigma$ in the example below):

```
cd  $REP_BASE_PROD/obs/isgri_gc
fcopy "isgri_srcl_res.fits[ISGR-SRCL-RES][DETSIG >= 7.0]" specat.fits
```

(Note, that there should be spaces around ">=").

Note that it is important to make the resulting catalog read only to avoid that it gets corrupted during the analysis:

```
 chmod -w specat.fits
```

Part of the resulting `specat.fits` catalog is shown in Figure 19. The same catalog will be used later for the lightcurve extraction.

In the GUI, set the **SCW2_cat_for_extract** parameter to point to `specat.fits` (use the **browse** button to get the full path) and press **Ok**, the window disappears and you are back to the main GUI page. There, press **Run** to launch the analysis.

7.2.1 Results of the Spectral Extraction

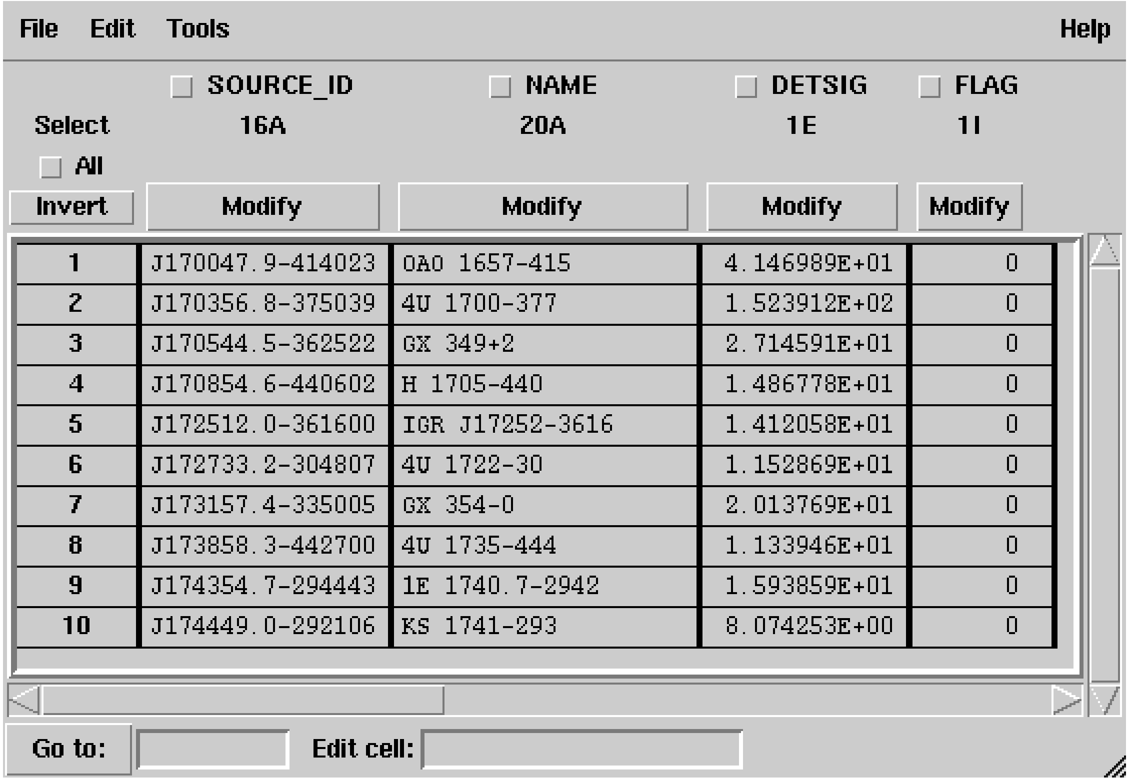
Spectral files are produced for each ScW:

```
scw/0051004X0010.001/isgri_spectrum.fits (X=1..5)
```

In these files you find the spectra of the desired sources plus the background spectrum. In the header of each data structure, the Name and ID of each source are given, and the first extension of the file scw/0051004X0010.001/isgri_spectrum.fits is an index, summarising the content of all the extensions. With the help of this index you can easily see which extension contains the spectrum of the source you are interested in.

To get an average spectrum with a better signal to noise ratio, one can sum up spectra of a source from different science windows. This can be done with the *spe_pick* tool. In the following example we create a joint spectrum of 4U 1700-377 from all the available science windows (included in the `og_ibis.fits` file).

```
cd $REP_BASE_PROD/obs/isgri_gc
spe_pick group="og_ibis.fits" source="4U 1700-377" rootname=4U1700
```



	SOURCE_ID 16A	NAME 20A	DETSIG 1E	FLAG 1I
1	J170047.9-414023	OAO 1657-415	4.146989E+01	0
2	J170356.8-375039	4U 1700-377	1.523912E+02	0
3	J170544.5-362522	GX 349+2	2.714591E+01	0
4	J170854.6-440602	H 1705-440	1.486778E+01	0
5	J172512.0-361600	IGR J17252-3616	1.412058E+01	0
6	J172733.2-304807	4U 1722-30	1.152869E+01	0
7	J173157.4-335005	GX 354-0	2.013769E+01	0
8	J173858.3-442700	4U 1735-444	1.133946E+01	0
9	J174354.7-294443	1E 1740.7-2942	1.593859E+01	0
10	J174449.0-292106	KS 1741-293	8.074253E+00	0

Figure 19: List of sources used for the spectral analysis

As a result two files with spectra of 4U 1700-377 will be created: `4U1700_sum_pha.fits` contains the final average spectrum of 4U 1700-377, while `4U1700_single_pha2.fits` stores all the original spectra of 4U 1700-377 that were averaged up. The corresponding Ancillary Response File (ARF) – which contains the energy dependence of the effective area of the instrument – is also produced and written to the `4U1700_sum_arf.fits` and `4U1700_single_arf2.fits` files. Note that the correct ARF is automatically chosen by the program from the list displayed in Table 3 and in case your observation spans more than one epoch, several ARFs are averaged up to produce the best ARF for your particular dataset.

7.2.2 Displaying the Results of the Spectral Extraction

The ISGRI systematics are typically of the order of 1%. We add this explicitly to the `4U1700_sum_pha.fits` file with the command below:

```
cd $REP_BASE_PROD/obs/isgri_gc
fparkey 0.01 4U1700_sum_pha.fits SYS_ERR  add=yes
```

One can now analyse the average spectrum with *XSpec* as:

```
xspec
    cpd /xw
    data  4U1700_sum_pha.fits
    setplot energy
    ign 300.-**
    model cutoffpl
    fit
    plot ldata delchi
```

Table 3: ARF instance number to be used

Period	Revolution interval	ARF instance XX isgr_arf_rsp_00XX.fits
1	1 - 63	31
2	64 - 135	32
3	136 - 254	33
4	255 - 370	34
5	371 - 481	35
6	482 - 635	36
7	636 - 750	37
8	751 - 800	38
9	801 - 900	39
10	901 - 1000	40
11	1001 - 1120	41
12	1121 - 1190	42
13	1191 - 1244	43
14	1245 - 1298	44
15	1299 - ...	45

```
hardcopy 4U1700_spe.ps
```

The resulting total spectrum of 4U 1700-377 saved to file `4U1700_spe.ps` is shown in Figure 20.

In the very same manner, you can sum up the spectra for all the sources of the input catalog, specified in the **SCW2_cat_for_extract** parameter[11].

Usage of *spe_pick* for spectra created in different observation groups is explained in Section 9.

7.3 Lightcurve Extraction

After you have done the spectral extraction, you may want to produce a lightcurve for the same sources. Stay in the working directory **$REP_BASE_PROD/obs/isgri_gc** and call the *ibis_science_analysis* script again. On the main GUI window (see Figure 12), change both Start Level and End Level to **LCR**, and press the **ISGRI SPE and LCR** button. The GUI shown in Figure 18 will appear.

Note that to extract the lightcurve for a source you need the Pixel Illuminated Fraction (PIF[12]) map. Such a map is created during the spectral step that we have just run. This means that at this stage you can extract the lightcurve only of the sources for which a spectrum has been created already. So leave **SCW2_cat_for_extract** to point to your `specat.fits`.

By default, for each source from the input catalog (`specat.fits` in our example), four lightcurves (in 20–40, 40–60, 60–100, and 100–200 keV energy bands) with 100 sec time bins will be created. You can change these values in the GUI.

The lightcurve extraction is performed by building shadowgrams for each time and energy bin. Hence this step is quite consuming in processing time and disk space. Note that due to CFITSIO limitations, **the product of number of energy bins by number of time bins in a ScW should be less than 250**.

Press **Ok**, the window will disappear and you are back to the main GUI page. Press **Run**.

[11]To read a given extension in XSPEC (version 12) you have to specify it in curly or squared brackets (depending on the spectral type): If you want to examine in XSPEC the spectra produced at scw level, you can load them as *e.g.,* `data scw/*/isgri_spectrum.fits\[3]` or as `data XXX_single_pha2.fits{3}`.

[12]PIF is a number between 0 and 1, which expresses the theoretical degree of illumination of each detector pixel for a given source in the sky.

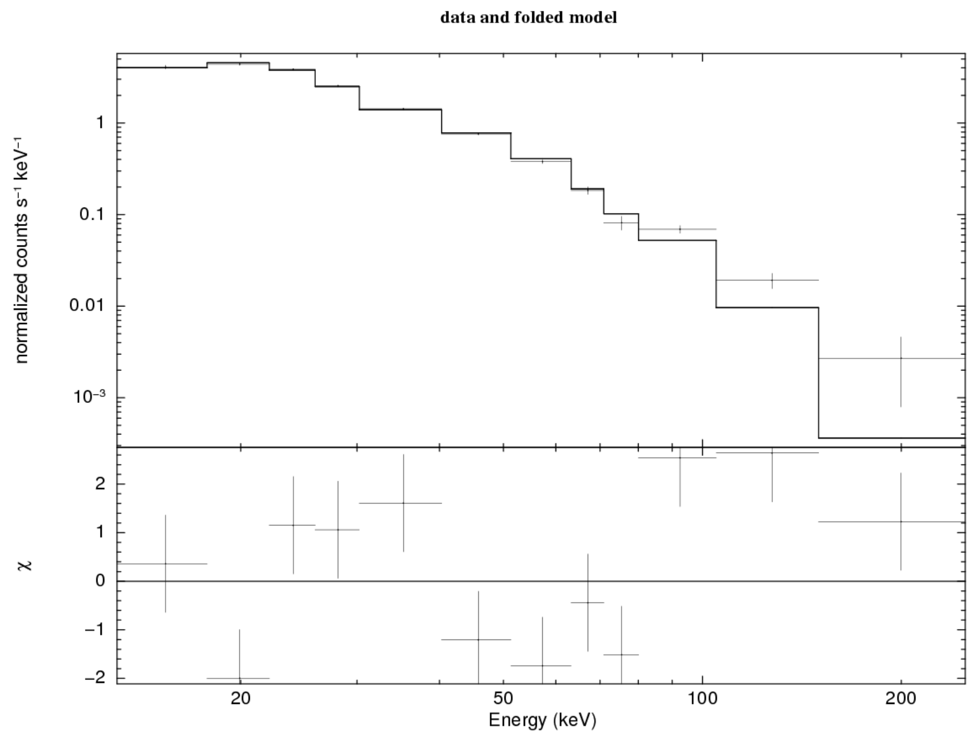


Figure 20: Total spectrum of 4U 1700-377 fitted with a cut-off powerlaw model.

7.3.1 Results of the Lightcurve Extraction

Lightcurves are produced for each science window:

```
scw/0051004X0010.001/isgri_lcr.fits (X=1..5)
```

To create a merged lightcurve in the 20–40 keV energy range (i.e. to store in one file all the available lightcurves of a given source) use *lc_pick*:

```
lc_pick source="4U 1700-377" attach=n \
        group=og_ibis.fits+1 lc=4U1700-377.fits emin="20" emax="40"
```

The result is written to **4U1700-377.fits**. It is possible to create more than one merged lightcurve at a time if you specify a list of desired minimum energies with the parameter `emin`. If you have more than one energy range with the same minimal energy, then you should use the parameter `lcselect` to define the unique energy band, otherwise the `lcselect` parameter can be omitted.

7.3.2 Displaying the Results of the Lightcurve Extraction

To easily see the source lightcurve, you can open the file `4U1700-377.fits` with *fv* and plot the column RATE with ERROR versus the column TIME, whereas to see the background lightcurve, you should plot column BACKV with error BACKE versus TIME. To display the resulting lightcurve it also convenient to use the *lcurve* program from the FTOOLS package:

```
lcurve
Number of time series for this task[1]
```

```
Ser. 1 filename +options (or @file of filenames +options)[file1] 4U1700-377.fits[2]
Name of the window file ('-' for default window)[-]
Newbin Time or negative rebinning[4.6692607009327] 100
Number of Newbins/Interval[10] 95    (take this number from the line above:
                                     Maximum Newbin No. 95)
Name of output file[default]
Do you want to plot your results?[yes]
Enter PGPLOT device[/XW]
PLT> hardcopy 4U1700-377_lc.ps/PS
PLT> quit
```

As a result, the 4U1700-377_lc.ps file was produced and is shown in Figure 21. Note that the count rates are already corrected for instrumental effects such as the off-axis transparency of the mask supporting structure.

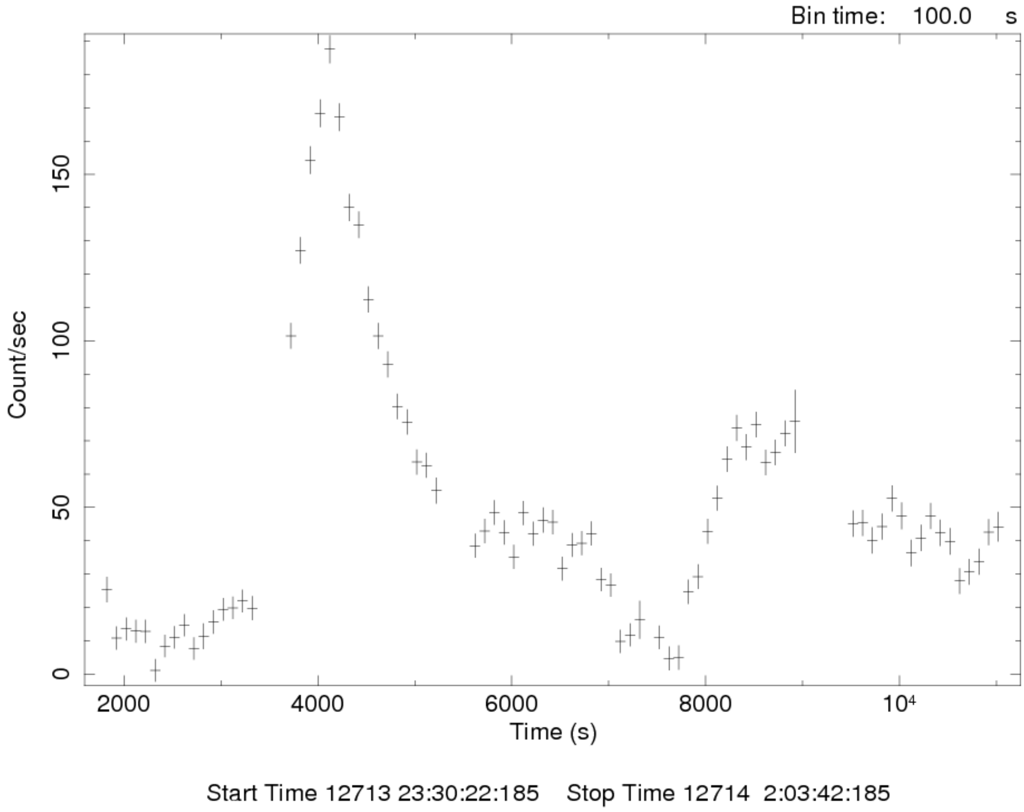


Figure 21: Lightcurve of 4U 1700-377 in the 20–40 keV energy range with 100 sec binning.

8 More on ISGRI relevant parameters

In the first part of the Cookbook (Section 7), the default values of the parameters were used. Now, it is time to discuss in more details the usage of the main parameters. The full list of parameters along with short explanations is given in Table 63, Appendix D.

8.1 How to choose the start and end level for the analysis. Intermediate levels.

Once you have specified `startLevel` and `endLevel`, all the steps listed in the `GENERAL_levelList` parameters between these two levels (both included) will be performed. For instance, in section 7.1 you used:

```
startLevel="COR" endLevel="IMA2"
GENERAL_levelList="COR,GTI,DEAD,BIN_I,BKG_I,CAT_I,IMA,IMA2,BIN_S,SPE,LCR,COMP,CLEAN"
```

The levels from COR to IMA2 (included) listed in `GENERAL_levelList` were performed.

You can choose what to run according to what kind of output you need. Basically:

- If you are interested only in imaging results set

  ```
  startLevel="COR" endLevel="IMA2"
  ```

  with the complete list of `GENERAL_levelList` given above, as done in section 7.1.

- If you have already a catalog of sources for which to extract the spectra (e.g.`specat.fits`[13]), do not have sources brighter than 600 mCrabs in the list (see warning at the end of this section), and you are not interested in the imaging results anymore you can skip all the imaging related levels and do:

  ```
  startLevel="COR" endLevel="SPE" GENERAL_levelList="COR,GTI,DEAD,BIN_S,SPE"
  ```

  Remember that if you do not do the imaging part then `isgri_srcl_res.fits` is not created! This file is the default one that is used for spectral extraction (7.2). Do not forget to specify the catalog you have built for this purpose setting the `SCW2_cat_for_extract` equal to `specat.fits`.

- Likewise, if you are interested only in lightcurve results and do not have sources brighter than 600 mCrabs in the list (see warning at the end of this section), you should create PIFs for all sources of interest with *ii_pif* program (see Section 9.11 for an example), and afterwards you may use:

  ```
  startLevel="COR" endLevel="LCR" GENERAL_levelList="COR,GTI,DEAD,BIN_S,LCR"
  ```

  again with **SCW2_cat_for_extract** equal to `specat.fits`.

- If you are familiar with the sources of your field (i.e. you have already run the imaging part, for instance) and you want to extract images, spectra and lighturves in one go, set `SCW2_cat_for_extract` equal to `specat.fits` and run your analysis with

  ```
  startLevel="COR" endLevel="LCR"
  ```

  and the complete list of levels in `GENERAL_levelList`.

Warning! If you skip the level BKG_I and do spectral extraction then the file with PIFs of strong sources is not done so no strong source subtraction is done in spectral extraction. If you really want to do step SPE or LCR without having done before the step BKG_I then you have two possibilities:

[13] `specat.fits` has to be of the ISGR-SRCL-RES type, meaning that it has to be created by the imaging step (with OBS1_DoPart2=1, since this file is created at mosaic stage). See section 7.2 to learn how to go from the imaging result file `isgri_srcl_res.fits` to your `specat.fits`, to avoid software crash.

1. there is no strong source in the FOV. In this case everything is OK
2. you have to run at least BKG_I and BIN_I (for example with just one energy band to save time)

8.2 Imaging

Source detection is an important and delicate process as the background and all the sources (especially the bright ones) in the field of view will have an impact on it. Because of this intrinsic complexity of the instrument, there are different options as far as source search and background correction are concerned. The choice of the best method to use for a given case may require a bit of iteration on your side.

In the next two paragraphs we describe the parameters involved in the source search, along with some practical advices. The single Science Window case is treated in Section 8.2.1, while the mosaic image case in Section 8.2.2. Comments on the background subtraction issue are given in Section 8.2.3 while a set of miscellanea on imaging parameters is given in Section 8.2.4.

8.2.1 How to choose the source search method in the Science Window analysis

Basically the idea is that you can either let the software look for the excesses in the field or you can tell the software to look for sources from a given catalog. This catalog can be either the one provided by ISDC (`CAT_refCat="$ISDC_REF_CAT"`), from which you can extract e.g. only the sources that have been detected by ISGRI ("`$ISDC_REF_CAT[ISGRI_FLAG>0]`"), or a catalog made on your own (see section 9.7 to create a catalog on your own). The number of sources that are detected in an image is important because detecting a source means also removing its ghosts from the images [14] (see section 7.1).

The source search method is defined with the parameters `OBS1_SearchMode` and `OBS1_ToSearch`. These parameters allow you the following possibilities:

- `OBS1_SearchMode` = 0
  This mode was developed for testing purposes and is not recommended for scientific use.
- `OBS1_SearchMode = 1`
  the software will look for all the sources of the input catalog that are in the field of view (regardless of their detection level, i.e. not using the OBS1_MinCatSouSnr parameter). If the source gives a positive detection, then the relevant results are saved in the `isgri_sky_res.fits` file, but see also discussion on OBS1_NegModels in Section 8.2.4.
  This mode can be useful in case your Science Window images are noisy and many fake new sources are detected. Searching only the sources from a given catalog will avoid having ghost cleaning for fake sources. This mode can be very time consuming if the observed field is "crowded", such as the Galactic Center region.
- `OBS1_SearchMode = 2`
  The software will look for the K brightest sources in the field, where `K = OBS1_ToSearch`.
  They can be either known sources (more significant than `OBS1_MinCatSouSnr`) or new sources (more significant than `OBS1_MinNewSouSnr`).
  This mode is useful if you are interested in having a first glance at your field, if you are interested in bright sources only, or if you run your analysis on large data sets. In fact, in these cases you would not like to spend most of your time collecting information on weak sources that are not detected in a single Science Window. The drawback of this method is that these undetected weak sources can produce ghosts in the mosaic image. We recommend you to start the analysis with `OBS1_SearchMode = 2` with a high `OBS1_ToSearch` value (around 50, especially for the Galactic plane or Centre crowded regions), see the sources that are detected and prepare your own catalog of detected sources (see section 9.7). Then launch again the analysis with `OBS1_SearchMode = 3` - described below - and give your catalog

[14] Note that the run time is proportional to the number of sources detected.

as input for the analysis (`CAT_refCat=usrcat.fits`).

WARNING: When using OBS1_SearchMode=2, the position of sources detected in the single science window and associated to catalog sources with ISGRI_FLAG=2 is assigned to the pixel centre, instead of being fixed to the catalog position as it should. This problem is not present with OBS1_SearchMode=1 or 3, since in this case the catalog position is used when no fitting is requested. (see also Sect.11).

- `OBS1_SearchMode = 3`
  The software will look for all the catalog sources (regardless of their detection level, i.e. not using the OBS1_MinCatSouSnr parameter) plus for K significant excesses more significant than `OBS1_MinNewSouSnr`, where `K = OBS1_ToSearch` up to a maximum of K sources. All these sources will be saved in the final **`isgri_sky_res.fits`** file if their detection significance is positive, but see also discussion on OBS1_NegModels in Section 8.2.4 . This mode is very efficient in cleaning the ghosts of all the sources in the field of view. It is of course more time consuming since a model is created and deconvolved for each given source.
  We suggest that you should use this mode either with your own catalog (built from the previous run with `OBS1_SearchMode = 2`, see above) or, if no personalised catalog is available, using the ISDC catalog with only the sources that have been detected by ISGRI (`$ISDC_REF_CAT[ISGRI_FLAG>0]`, see above). Any bright source out of the catalog (new or forgotten!) will be detected as one of the K new sources.

8.2.2 Parameters related to the mosaic step

The `OBS1_SearchMode` parameter concerns only images at a Science Window level. In the mosaic image the software always looks for `K=OBS1_ToSearch` sources (more significant than `OBS1_MinCatSouSnr` for catalog sources, or `OBS1_MinNewSouSnr` for the new sources), regardless the real number of sources in the field of view. Thus, if you are working with a crowded part of the sky (mainly Galactic plane and Centre) you should use a high OBS1_ToSearch number (at least 50).

There are two ways to calculate the source flux in the mosaic given the original Science Window information:

- `OBS1_PixSpread = 0`
  The whole input (Science Window) pixel count is put into one output (mosaic) pixel, no "spread" is done. This method optimises the flux and signal-to-noise ratio (SNR) evaluation but can sometimes give some undesired effects such as double source peaks.

- `OBS1_PixSpread = 1` (default value)
  The input pixel count is spread between different output map pixels. This method is better for the source position estimate but is less accurate than the previous method for source flux estimation. SNR in the new spread mosaic will be lower than in the previous case. The source flux in this case is reduced with respect to the weighted mean of individual science windows (or mosaic obtained with `OBS1_PixSpread = 0`). It is not feasible to give the general estimate of this flux reduction, as it depends on image binning conditions and is different for each source position. The displacement of the center of the mosaic could be enough to change source flux. Thus, the mosaic created with `OBS1_PixSpread = 1` is not suitable for flux estimation.

It is possible to decide whether you want the mosaic to be created along with the Science Window image reconstruction, or if you prefer it to be created later on. The mosaic production is controlled by the `OBS1_DoPart2` parameter:

- `OBS1_DoPart2 = 0`
  The mosaic step is not performed and the imaging part ends after the Science Window based part.

- `OBS1_DoPart2 = 1` (default value)
  The mosaic is done after the Science Window image reconstruction, in the same run.

- `OBS1_DoPart2 = 2`
  The mosaic is performed for images obtained from different, already existing, runs (see 9.4.1).

8.2.3 Background Subtraction

In the current OSA you have background maps provided by the ISGRI team. These maps are created on the base of observations containing only faint sources in their fields of view.

Background subtraction can be tuned with the help of the `SCW1_BKG_I_isgrBkgDol` parameter. If it is left empty then the background map is taken automatically from the IC files. You can, in principle, provide your own background map (which is e.g. specially calculated for the time period of the analyzed observation). As a general guideline, for short time scale (e.g. 1 Science Window), the background features should be less than the statistical uncertainties so that background removal is not expected to change the result by much. On longer time scales (mosaic image), things can be different, especially in the case of a Staring observation in which background features can accumulate and become important. To have no background subtraction, set `SCW1_BKG_I_isgrBkgDol=''-''`.

If a bright source is present in the IBIS field of view, then it can change the relative normalization factor for the background maps. To avoid this you should provide to the script a list of bright sources in the FOV. This can be done with the parameter `brSrcDOL`. By default all sources previously detected by ISGRI (`ISGRI_FLAG2==5`) with catalog flux in the 20–60 keV energy band brighter than 100 count/s (∼600 mCrab) (`ISGR_FLUX_1>100`) are taken into account (cf. [1]). You can check if addition of other bright sources (e.g. the ones seen in the mosaic of your observation) to the bright source catalog improves the quality of the background subtraction in your particular observation. Apart from the persistent bright sources, it is also recommended to include in the bright source catalog bright transients which are visible in the FOV and which were active during the period of observation. Addition of a source XX to the bright source catalog can be done by changing the filtering of the ISDC reference catalog: `brSrcDOL="$ISDC_REF_CAT[ISGRI_FLAG2==5&&ISGR_FLUX_1>100||NAME=='XX']"` The number of bright sources which could affect the overall normalization of the detector background count rate can be very large in a crowded field, like e.g. the Galactic Center region, considered above. In this case the number of detector pixels ingored during the calculation of the background normalization can become very large. This can result in a worsening of the quality of background subtraction. To avoid this, one can try two options. First possibility is to leave the bright source catalog empty, `brSrcDOL=""`, so that the entire "collective" contribution of all the bright sources, which is almost the same in all detector modules, will be taken into account. Next, one can specify if the normalization of the background is calculated using the entire detector, or on module-by-module basis. This can be done by changing the value of parameter `SCW1_BKG_I_method_cor` between `1` (default, the entire detector) and `2` (on module-by-module basis). In addition, one can try a combination of these two options (both give the empty bright source catalog and force the calculation of the normalization of background using the entire detector).

8.2.4 Miscellaneous on Imaging

- With the parameter `OBS1_ExtenType` (to find it on the GUI pages press the hidden button, and go to the IMA page) you can choose whether to keep the Residual and/or the true Exposure image in the `isgri_sky_ima.fits` file. With `OBS1_ExtenType`="0", 4 maps will be created per energy range (Intensity, Variance, Significance and Residuals). With `OBS1_ExtenType` = 1 true exposure maps will be written instead of residual images at the 4*n-th extension of `isgri_sky_ima.fits`. The new default value, "2" (instead of "0" before OSA 10), will add the true Exposure maps in an additional extension. `OBS1_ExtenType` = 3 will add single true Exposure map as the last extension. Please note that the change of default value (from "0" to "2") affects also the mosaic images as instead of an on-time map in the image mosaic file (`isgri_mosa_ima.fits`) one now gets an effective exposure map which has much lower values, because of the half-opening of the coded-mask and various vignetting effects that are here taken into account.
- If you are interested in running the analysis in more than 10 energy bands please see Section 9.5.
- You can decide if you want to let the software find the best fit position of a source (in the vicinity of the one in the catalog, default `OBS1_SouFit=0`), or if you would prefer the catalog position (for all sources from the input catalog!) to be used for the flux determination (`OBS1_SouFit=1`). If `OBS1_SouFit=1`, then the position of a new source is always attributed to the pixel center, and for catalog sources

the catalog position is used for SearchMode=1,3, or the pixel center for SearchMode=2. Note that `OBS1_SouFit=1` fixes all the source positions only in the individual Science Window analysis. In the mosaic, all source positions are always fitted regardless of the value of `OBS1_SouFit`. These fitted values are written to RA_FIN and DEC_FIN columns of isgri-srcl-res.fits, and are later used in spectral and lightcurve extraction steps as a source position. To fix the positions of only several sources, set for them ISGRI_FLAG=2 in the input catalog. For the weak sources with well-known positions it is recommended to use the catalog position (`ISGRI_FLAG=2`), whereas for bright sources, whose position is determined by the software with an accuracy better than 15″, it is recommended to let the software to fit the position of the sources (`ISGRI_FLAG=1`).

WARNING: When using OBS1_SouFit=1 with OBS1_SearchMode=2, not only the position of new sources but also that of cataloged ones is assigned to the pixel centre, instead of being fixed to the catalog position. It is therefore strongly recommended to avoid this combination of parameters. This problem is not present with OBS1_SearchMode=1 or 3, since in this case the catalog position is used when no fitting is requested (see also Sect.11).

- For `OBS1_SearchMode` = 1 or 3 (where you are forcing a catalog extraction) ghosts can be cleaned even from the known sources that happen to have a negative peak. This is done using `OBS1_NegModels = 1` while with `OBS1_NegModels` empty or equal to "0" (the default) the deconvolved images will be cleaned from positive source ghosts only.

8.3 Spectral and Timing Analysis

8.3.1 Spectral Energy Binning

With the help of the `IBIS_SI_inEnergyValues` parameter you can specify the desired energy bins. Apart from specifying the energy bins, you should also rebin the responce matrix, see Section 9.5 for the detailed discussion on how to do this. The rebinned response matrix should then be linked to the spectrum you produce. This can be done by setting the parameter
SCW2_ISPE_idx_isgrResp="your_rebinned_matrix.fits". This parameter defines the name of the response matrix that would be added to the header of the spectrum file. **Warning:**This parameter is a hidden one. Thus you should either provide the value of this parameter during the script launch

```
ibis_science_analysis SCW2_ISPE_idx_isgrResp="your_rebinned_matrix.fits"
```

or press "hidden"-button and set this parameter on the SPE page.

8.3.2 Background Subtraction

In spectral and timing analysis you can choose (with the help of the parameter `SCW2_BKG_I_isgrBkgDol`) whether you want to subtract the instrumental background or not, similarly to the imaging case. We recommend you to use the option that gives the best results at the imaging level (smoother image, no clear patterns etc). See 8.2.3 for more information.

8.3.3 Input catalog

The single Science Window and mosaic imaging results are merged in the file `isgri_srcl_res.fits`. By default a spectrum and a lightcurve will be created for all sources listed in this file. Spectral/lightcurve extraction is time consuming and we recommend you to create an input catalog for spectral and timing extraction, as discussed in Section 7.2. The spectrum and lightcurve extraction tools use the source position saved in the RA_FIN and DEC_FIN columns. These are the values computed during the imaging step, thus in case you want to use catalog source positions you have to modify them manually. In case you prefer

the catalog positions for all catalog sources, you can remove RA_FIN and DEC_FIN columns (RA_OBJ and DEC_OBJ are used then) with the following commands:

```
chmod +w specat.fits
fdelcol infile=specat.fits+1 colname="RA_FIN" confirm=no proceed=yes
fdelcol infile=specat.fits+1 colname="DEC_FIN" confirm=no proceed=yes
chmod -w specat.fits
```

Important: make the file with the catalog **read only**, otherwise it may be corrupted in the course of the analysis!

9 Useful recipes for the ISGRI data analysis

In this Section we give a number of recipes that can be useful in the analysis of ISGRI data.

9.1 Rerunning the Analysis

Read this if you would like to redo part of your analysis, e.g. if your run has crashed, or if you want to change some parameters.

In case you want to re-run the analysis with different parameters, run `og_create` but this time with a different "ogid" parameter. This will create a new tree under `obs/ogid` where all the new results will be stored. If the pipeline has crashed[15], in general it is safer to restart your analysis from scratch removing the `obs/ogid` directory and restarting from the `og_create` step.

In any case, we give below a set of recipes that can be useful.

Because of the group concept you cannot just delete the result you do not like and restart the pipeline. All results that were produced in the course of the analysis are linked to the group, and should be detached before you relaunch the script. To do this you can use the *og_clean* program, that will clean an Observation Group up to the level specified with parameter `endLevel`. All data structures with a level equal or prior to `endLevel` will be kept, while the data structure with a later level will be erased. For example, to run the spectral extraction (SPE level) you should clean from the group whatever comes after the BIN_S level, as this is the level immediately preceeding the spectral one (see Fig.12).

```
og_clean ogDOL="og_ibis.fits" endLevel="BIN_S"
```

If og_clean fails it could be due to the fact that the group was corrupted. You should try to fix it with *dal_clean* program

```
dal_clean inDOL="og_ibis.fits" checkExt="1" backPtrs="1" checkSum="1"
```

and launch *og_clean* only afterwards.

Unfortunately the current version of *og_clean* is very slow. In some cases, if you know exactly which data structure has to be detached it would be much faster to launch *dal_detach*. For example if you run your analysis till SPE, or LCR level, and would like to produce a mosaic image afterwards, you do not have to clean the group, deleting all your results, but just have to detach ISGRI-SRCL-RES data structure (note that with the option "delete=y" all files with this data structure will be deleted):

```
dal_detach og_ibis.fits\ pattern=ISGR-SRCL-RES delete=y
```

9.2 Make your own Good Time Intervals

Read this if you are interested in selecting photons arrived at a particular time period (e.g. in analysis of flares, or for phase resolved spectroscopy).

You should define Good Time Intervals (GTIs) with the help of the *gti_user* program. To create a GTI for IBIS starting on IJD 1322.68 and lasting 1 minute (do not forget to convert it into days!) give the command:

```
gti_user gti=user_gti.fits begin=1322.68  end= \
             length=6.944E-4  group=og_ibis.fits\ unit='day'
```

[15] If any executable crashes then it terminates with non zero status. The meaning of the status value can be found at http://www.isdc.unige.ch/integral/analysis#Errors

Note that the parameter `unit` governs simultaneously `begin, end, length` and `step` parameters. If `unit`='day', then `begin` and `end` are considered to be absolute values in IJD. If `unit`='sec' then, `begin` and `end` are taken relative to "tstart" (See the full list of gti_user parameters in [1]).

It is also possible to define the GTIs of the fixed length (parameter `length`), separated by the given step (parameter `step`). In the example below 6 GTIs of 0.1 day length, beginning at time IJD 1092.5. and separated by 0.2 days are defined.

```
gti_user gti=user_gti_repeat.fits begin=1092.5 end= \
length=0.1 repeat=5 step=0.2 group=mygroup.fits\
```

Then in the main page of the GUI, Fig. 12, you should set `SCW1_GTI_gtiUserI="user_gti.fits"`, and specify the time format that was used in this file, in the example above `SCW1_GTI_TimeFormat="IJD"`. More details on *gti_user* are in the Data Analysis section of [1].

9.3 Usage of the predefined Bad Time Intervals

Read this to know how to make proper selection of the science windows that should be used in your analysis.

Different things could happen to the instrument during a particular science window. To inform the user on any unusual things that he should be aware of, the list of time intervals which had some anomalies (bad time intervals) has been created and is kept in the latest $REP_BASE_PROD/ic/ibis/lim/isgr_gnrl_bti_*.fits file.

In this list you find the following 7 categories:

- IBIS_CONFIGURATION – is to flag the change of configuration. Its time interval is short, just the duration of the switch of parameters. Column COMMENTS gives changes on the 4 main parameters (given in file instr_settings.txt), by order of importance:
  - rise-time selection
  - VETO configuration
  - low threshold adjustment
  - COMPTON window

  It is recommended not to use in your analysis any science window with such a problem.
- ISGRI_RISE_TIME – indicates that on-board cut on rise-time is too low, so we don't have arf and rmf for this case.
- BELT_CROSSING – indicates that belts are seen in VETO or ISGRI count-rates.
- SOLAR_FLARE – indicates periods where a strong solar affected part of the data.
- VETO_PROBLEM – contains periods when VETO had a strange behaviour with a count-rate much lower (or higher) than expected. If it is lower, then less ISGRI events were killed, and hence science count-rate is higher. Thus, user can analyze these data but must not rely on the flux (dead-time can be wrong). Also 2 cases of "voltage breakdown" are added to this category.
- IBIS_BOOT – indicates that IBIS has been restarted from OFF state unexpectedly.
- MISCELLANEOUS – For the moment there are cases where we have problems with the time values, and all others are related to the drop of PICsIT counters.

For the most conservative data analysis use the following value of the SCW1_GTI_BTI_Names parameter:

```
SCW1_GTI_BTI_Names="IBIS_CONFIGURATION ISGRI_RISE_TIME BELT_CROSSING SOLAR_FLARE VETO_PROBLEM IBIS_BOOT MISCELLANEOUS"
```

9.4 Combining results from different observation groups

Read this if you have a set of science windows belonging to different runs for which you have already built images, spectra or lightcurves and want to combine the results.

Section 9.4.1 explains how to combine all the existing images in a final mosaic, while Section 9.4.2 shows how to merge different lightcurves and spectra.

9.4.1 Creating a mosaic from different observation groups

Suppose you want to analyse 5 Science Windows and are not interested in the final mosaic. You create the group with `og_create` and then you launch the analysis till the imaging step but without the mosaic step (i.e. from COR till IMA with `OBS1_DoPart2 = 0`, see 8.2.2). An image is created per Science Window but you do not have the overall final mosaic. If you then change your mind and decide that you want the mosaic, all you have to do is move in the working directory $REP_BASE_PROD/obs/xxx and relaunch the `ibis_science_analysis` command with `startLevel="IMA"`, `endLevel="IMA"` and `OBS1_DoPart2=1`.
This is relatively simple because all the Science Windows belong to the same group, so combining the results is trivial. But if this is not the case, i.e. if you have run different sets of analysis, each one with its own `og_create` command, then you need to make some intermediate steps [16]. You basically need to create a file that points to all the Science Windows you want to co-add. We call this file an "index". To create an index make a list of the Science Window groups you want to combine and save it as, e.g., **dols.txt**. To ensure a proper work of the software give the full path, i.e. your file should look like

/WORKING-DIR/obs/GROUP1/scw/011901070010.001/swg_ibis.fits
/WORKING-DIR/obs/GROUP1/scw/011901080010.001/swg_ibis.fits
/WORKING-DIR/obs/GROUP2/scw/012000360010.001/swg_ibis.fits
/WORKING-DIR/obs/GROUP2/scw/012000370010.001/swg_ibis.fits

The first 2 files belong to a run with the `og_create` parameter "ogid" equal to "GROUP1" while the latter two to a run with "ogid" equal to "GROUP2". /WORKING-DIR/ has to be the extensive name of $REP_BASE_PROD. **WARNING:** make sure that `obs/*/scw/*.001/isgri_sky_ima.fits` files exist in all the Science Windows you mention, otherwise the merging will not work. Then give the command that actually builds the fits file from the ASCII file:

```
cp dols.txt $REP_BASE_PROD/obs/GROUP1/
cd $REP_BASE_PROD/obs/GROUP1/
txt2idx element="dols.txt" index="index_comb.fits"
```

The file "index_comb.fits" is created and you can look at it with **fv**. In the first extension you have 4 rows, each row has the link to a given swg_ibis.fits file. What you need to do now it to let the software know that it has to use this particular set of Science Windows for the analysis. You do this by pointing **og_ibis.fits** to this file. This has to be done because no matter what level of the scientific analysis you are performing, the software will analyse the Science Windows pointed to by **og_ibis.fits**. Replace the first row of og_ibis.fits with:

```
cd $REP_BASE_PROD/obs/GROUP1/
ftedit og_ibis.fits MEMBER_LOCATION 1 index_comb.fits
fdelrow og_ibis.fits+1 2 1 N Y
rm isgri_catalog.fits
```

With the `fdelrow` command you detach from the first extension of **og_ibis.fits** row number "2" (= the first row to be deleted, number of rows to be deleted = "1"), you were shown no keyword values, "N", and

[16] Please note that the images that you intend to merge MUST have the same energy boundaries i.e. all the ScW maps must have been analysed in the same way. You cannot merge a 20–40 keV map with a 30–50 keV one.

you agreed to proceed ,“Y”. This second row contained the catalog that was created in previous run during CAT_I step, and contain only sources that were in the field of view of science windows from GROUP1. To have a common catalog for GROUP1 and GROUP2 it would be necessary to rerun CAT_I step for the group referring to all science windows of interest.

You may check with **fv** that actually the first row of og_ibis.fits is indeed ”index_comb.fits”.

Before running the mosaic step please be aware that:

- If you had previously run the mosaic step you will see that **og_ibis.fits** points to the mosaic output (after you have deleted a reference to isgri_catalog.fits rows 2 to 4 of the first extension are isgri_srcl_res.fits, isgri_mosa_ima.fits and isgri_mosa_res.fits). This will interfere with the new mosaic you are about to launch, thus you have to detach these former mosaic results from the group and delete (or rename) them:

```
cd $REP_BASE_PROD/obs/GROUP1
fdelrow og_ibis.fits+1 2 3 N Y
rm isgri_mosa_ima.fits
rm isgri_mosa_res.fits
rm isgri_srcl_res.fits
```

- If you had previously run the spectral and lightcurve steps you have to detach the ISGRI-SRCL-RES data structure from all your Science Window groups (swg_ibis.fits). That is to each Science Window ($scw) group located in the $REP_BASE_PROD/obs/$dir directory, you should do the following command:

```
cd $REP_BASE_PROD/obs
dal_detach object="$dir/scw/$scw.001/swg_ibis.fits" pattern=ISGR-SRCL-RES delete=n
```

At this point you are ready to launch the analysis and to create a joint mosaic:

```
cd $REP_BASE_PROD/obs/GROUP1
ibis_science_analysis ogDOL="og_ibis.fits" \
startLevel="CAT_I" endLevel="IMA"  OBS1_DoPart2=2
```

See 8.2.2 for a description of the main parameters of the mosaic step. The above command is valid in case you have just run the ScW analysis part so that you indeed keep the same energy boundaries. Otherwise, if your parameter file has changed in between, you should add in the above the definition of `IBIS_II_ChanNum`, `IBIS_II_E_band_min` and `IBIS_II_E_band_max` according to the boundaries of the single ScW maps you want to merge.

9.4.2 Combining spectra and lightcurves from different observation groups

In Sections 7.2 and 7.3 you have seen how to merge lightcurves and spectra from different Science Windows belonging to the *same* group. In that case the file **og_ibis.fits** points to all the Science Windows and it is possible to launch the collecting/merging tools *lc_pick* and *spe_pick* directly on the group (group=og_ibis.fits+1).

If you have Science Windows belonging to *different* groups, you need an intermediate step. Basically you need to create a file (an index) that points to all the Science Windows you want to co-add, similarly to the case seen in section 9.4.1. Then this file will be given as input via the ”group” parameter at the place of **og_ibis.fits**.

To create the index make a list of the Science Window groups you want to combine and save it as, e.g., **dols.txt** under $REP_BASE_PROD/obs/. To ensure a proper work of the software give the full path, i.e. your file should look like

/WORKING-DIR/obs/GROUP1/scw/011901070010.001/swg_ibis.fits
/WORKING-DIR/obs/GROUP1/scw/011901080010.001/swg_ibis.fits
/WORKING-DIR/obs/GROUP2/scw/012000360010.001/swg_ibis.fits
/WORKING-DIR/obs/GROUP2/scw/012000370010.001/swg_ibis.fits

The first 2 files belong to a run with the `og_create` parameter "ogid" equal to "GROUP1" while the latter two to a run with "ogid" equal to "GROUP2" /WORKING-DIR/ has to be the extensive name of $REP_BASE_PROD. **WARNING:** make sure that the lightcurve and spectra result files exist for each Science Window you want to co-add (isgri_lcr.fits and isgri_spectrum.fits files).

Then give the command that actually builds the fits file from the ASCII file:

```
cd $REP_BASE_PROD/obs/
txt2idx element="dols.txt" index="index_comb_2.fits"
```

Then run *lc_pick* and *spe_pick* putting the parameter "group" equal to "index_comb_2.fits":

```
cd $REP_BASE_PROD/obs/
lc_pick source='GRS 1758-258' attach=n \
group=index_comb_2.fits+1 lc=GRS1758.lc.fits emin="20"  \
emax=40 instrument="ISGRI"
```

GRS1758.lc.fits contains the merged lightcurve of GRS 1758-258 in the 20–40 keV band (energy range that of course has to exist in the isgri_lcr.fits original files!).

```
cd $REP_BASE_PROD/obs
spe_pick group="index_comb_2.fits"\
               source="GRS 1758-258" rootname="GRS1758"  instrument="ISGRI"
```

GRS1758_sum_pha.fits (the combined spectrum of GRS 1758-258) and GRS1758_single_pha2.fits (a file with the four spectra of the initial four Science Windows collected together) are created. *spe_pick* also creates an ARF, appropriate for your particular dataset. This ARF is written to GRS1758_sum_arf.fits and GRS1758_single_pha2.fits files. The names of the response and ancrfile are inserted in the keyword of the final files, so that they are automatically recognised by XSPEC.

9.5 Rebinning the Response Matrix

Read this if you want to use a spectral binning different from the default one. You will also learn how to rebin the response matrix to extract images in more than 10 energy ranges.

The file `$REP_BASE_PROD/ic/ibis/rsp/isgr_rmf_grp_0025.fits` contains the latest full response matrix with 2048 channels. With the help of the FTOOLS program **rbnrmf** you can rebin this matrix according to your needs. The default rebinned matrix is `$REP_BASE_PROD/ic/ibis/rsp/isgr_rmf_grp_0027.fits` and has 13 energy bins in its third extension. The optimum way to rebin the matrix depends on the source and on what you are interested to study, so there is not a general best way to rebin it. Nevertheless, it is reasonable not to spend time on a fine energy binning where the source is too weak and background contaminated as well as it is better to avoid too narrow energy bins for weak sources: extracting a spectrum in wider energy bins from the very beginning is better than extracting it in too fine bins, being background contaminated, and then rebinning the spectrum afterwards.

To use **rbnrmf** you should create an ASCII file with three columns, representing minimal and maximal channels, and the compression factor. In the example below we create the file `new_bin.txt` in order to apply

a compression factor of 16 to channels 10 – 409 (17.7 – 209.26 keV), and ignore all the others. This will lead to [(maximal_channel-minimal_channel+1)/compression factor] number of final bins, i.e. in this example to 25 bins.

```
 0   9   -1
 10  409  16
 410 2047 -1
```

To rebin the matrix give the command:

```
rbnrmf infile="$REP_BASE_PROD/ic/ibis/rsp/isgr_rmf_grp_0025.fits" \
outfile="new-rmf.fits" binfile="new_bin.txt"
```

As a result of this, you now have the rebinned response matrix `new-rmf.fits`. With the help of **fv** you can see that the rebinned matrix is saved in the third extension of the file. For the successful analysis you should make new rebinned matrix read-only:

```
chmod -w new-rmf.fits
```

To tell the software to use this newly created matrix set in your analysis
`IBIS_SI_inEnergyValues="/PATH-TO-THE-FILE/new-rmf.fits[3]"`
otherwise the default 13 channels will be used (corresponding to an empty value of `IBIS_SI_inEnergyValues`).

To have the correct name of the response matrix in the header of the spectrum file you should set SCW2_ISPE_idx_isgrResp= rmf.fits". **Warning:**This parameter is hidden. Thus you should either provide the value of this parameter during the script launch

```
ibis_science_analysis SCW2_ISPE_idx_isgrResp="new-rmf.fits"
```

or press "hidden"-button and set this parameter on the SPE page.

9.5.1 Extracting images in more than 10 energy ranges

If you want to create images in more than ten energy bands, you have to create a rebinned matrix **new-rmf.fits** as shown in the text just above. Then, to create images in the energy ranges defined in the third extension of this file set
`IBIS_II_ChanNum=-1`
`IBIS_II_inEnergyValues="/PATH-TO-THE-FILE/new-rmf.fits[3]"`.
The first parameter instructs the program to use the energy boundaries specified in the second parameter.

9.6 Some tricks on saving disk space and CPU time

To gain time and space it is possible to create the rebinned correction and background maps rebinned_*.fits only once and then give them as an input for the rest.

Analyse one science window from COR to SPE, copy the maps somewhere and make them read only. Then launch the analysis for other science windows specifying the following parameters in your *ibis_science_analysis* call:

```
rebinned_corrDol_ima="/your_path/rebinned_corr_ima.fits"
rebinned_corrDol_spe="/your_path/rebinned_corr_spe.fits"
rebinned_backDol_ima="/your_path/rebinned_back_ima.fits"
```

```
rebinned_backDol_spe="/your_path/rebinned_back_spe.fits"
rebinned_unifDol_ima="/your_path/rebinned_unif_ima.fits"
rebinned_unifDol_spe="/your_path/rebinned_unif_spe.fits"
```

IMPORTANT: rebinned maps are created for a given set of energies only, so these should remain the same only as long as you are not changing energy ranges in your IMA and SPE analysis. If you want to change IMA energy bins or spectral response energy boundaries, these maps have to be created again.

9.7 Create your own catalog

Read this if you are familiar with the sources in your field of view and you want to build your own catalog for the Imaging step.

As already discussed in Section 8.2.1, during the imaging step (IMA) it can be very useful to use your own input catalog. The easiest way to build it is to modify the general catalog with the help of the `fcopy` program of FTOOLS. For example, if you want to create a catalog (`usrcat.fits`) with only three sources, Crab, 3C111, and XPer do the following:

```
cd $REP_BASE_PROD
fcopy "$ISDC_REF_CAT[NAME == 'Crab'|| NAME == '3C 111'|| NAME == 'X Per']" usrcat.fits
chmod -w  usrcat.fits
```

If you want to add a new source, not specified in the general reference catalog, add a line to `usrcat.fits` with the help of the **fv** tool, and fill in the Source_ID, NAME, RA_OBJ and DEC_OBJ columns. As Source_ID you can choose whatever you like provided that this identifier is unique.

NOTE: `usrcat.fits` keeps the same format of the general reference catalog from which it was built, i.e. it has the extension of the type "GNRL-REFR-CAT". This is the correct extension to be used as input for the Imaging step. In Section 7.2 you have been shown how to build a catalogue for the lightcurve (LCR) and spectral (SPE) step. In that case the catalogue you obtained, `specat.fits`, was of the type of "ISGR-SRCL-RES" which is the correct format for the LCR and SPE steps. The two formats cannot be switched, meaning that you cannot use a "GNRL-REFR-CAT" type catalog for SPE and LCR or an "ISGR-SRCL-RES" one for IMA.

9.8 Alternative Spectral Extraction from the Mosaic

Read this if you want to extract the average spectrum of a source from a set of mosaic images in different energy bands.

It is possible to extract the spectrum of a source from a set of mosaic images (in several large energy bands). In general this method gives results similar to the Science Window average spectrum you obtain from *spe_pick* (see Sections 7.2 and 9.4.2). Once you have the spectrum, you will need the corresponding rebinned matrix to be able to analyse it in XSPEC. We suggest to build such a matrix before launching OSA as shown in Section 9.5 and then extract images with the use of this matrix as shown in Section 9.5.1. See Sections 7.1 and 8.2.2 to learn how to create mosaic images (for a better flux estimate use OBS1_PixSpread=0).

Once you have the set of mosaic images in the desired energy bands, to extract the spectrum from the mosaics (attached to the og_ibis.fits file!) at the position of 4U 1700-377 (ra=255.9865, dec=-37.84414) use *mosaic_spec*:

```
cd $REP_BASE_PROD/obs/xxx
mosaic_spec DOL_inp="og_ibis.fits" DOL_out="og_ibis.fits"\
EXTNAME="ISGR-MOSA-IMA" DOL_spec="4U1700_mosa_pha.fits(ISGR-PHA1-SPE.tpl)"\
ra=255.9865 dec=-37.84414 size=4
```

Note that if the significance of the point is less than five, then *mosaic_spec* will assign a non-zero value to its quality, and the point will be not used by XSPEC. If you are interested to see this point do change its quality value to zero manually. You find more details on *mosaic_spec* in Section 12.12.1.

Please, remember, that different ARFs should apply to spectra extracted from the data of different observation periods (see Table 3). If the mosaic, from which you have extracted spectra, contains the data from more than one observation period, none of the pre-defined ARFs can be used with the spectra generated with *mosaic_spec*. Instead, a special ARF, averaged proportionally to the exposures from different periods should be generated. The simplest way to do this is to run the standard ScW by ScW spectral extraction for your source (even if the source is not detected in individual ScWs) and then use the *spe_pick* command (see Section 9) which, among others, will produce the exposure-weighted average ARF for your particular data set.

9.9 Barycentrisation

Read this if you want to apply barycentrisation to an extracted lightcurve.

The tool making such a correction is called *barycent*. Note that it re-writes the input file, so it may be worth to copy the original file first. Below we show an example for 4U 1700-377, Science Window number 011800900010 and observation group identification ("ogid" parameter of og_create) equal to "OGID":

```
cd $REP_BASE_PROD/obs/OGID
cp scw/011800900010.001/isgri_lcr.fits isgri_lcr_011800900010_bar.fits
barycent inCOL=TIME outCOL=TIME \
outDOL="isgri_lcr_011800900010_bar.fits[2]" \
inDOL='' auxDOL="../../aux/adp/0118.001/orbit_historic.fits" \
raOBJ=255.9865 decOBJ=-37.84414
```

In the example we have applied barycentrisation to the first energy range of 4U 1700-377 that happens to be in the extension number 2 ("[2]"). Thus the extension you give as "outDOL" depends on the source and energy range you need to correct. The overview of the content of each extension is in the input file isgri_lcr.fits, i.e. the first extension (`GROUPING`).

Note that *barycent* tool does not use group concept, and is used directly with the lightcurve. But for it successful work **the REF_BASE_PROD variable should be set** and point to the place in which there is an ic directory, e.g. `directory_of_ic_files_installation_/ic`.

See more details on *barycent* tool in the Data Analysis section of [1].

9.10 Alternative Timing Analysis

Read this if you are interested in extracting lightcurves on smaller time bins (i.e. down to about 0.1 sec) than the ones allowed by the standard tool.

9.10.1 ii_light

The standard lightcurve extraction tool (*ii_lc_extract*) called within the general analysis script builds shadowgrams for each requested time and energy bin. Thus this program is quite time consuming and it is not recommended to use it with time bin less than about 60 seconds. Besides, be aware that the higher the number of total bins is, the higher is the time and space needed.

The *ii_light* tool (not called in the analysis script but available as a stand alone tool) uses a different algorithm: the lightcurve is extracted not via the shadowgram creation but with the use of the Pixel Illuminated Fraction (PIF). This allows to extract lightcurves up to a time bin of about 0.1 sec. **This is not the official**

lightcurve extraction tool and should be used mainly to check relative variability of bright sources within a given Science Window, rather than for a long term absolute flux estimate. Note that to be able to run *ii_light* you need to have ROOT available.

9.10.2 Run `ii_light`

To be able to use *ii_light* you have to run the standard analysis until the SPE level (`startLevel="COR" endLevel="SPE"`). **You need to run the IMA level** with the *same* energy ranges in which you are going to extract the lightcurves. This is due to the fact that the IMA step will produce correction and background maps (rebinned_corr_ima.fits and rebinned_back_ima.fits) that are energy dependent and that will be put as input to *ii_light*. Alternatively you can use *ii_map_rebin* to produce these maps in the desired energy bands. **You need to run the SPE level** because during this level the PIF, needed as input to *ii_light*, is created. Alternatively you can create PIF with *ii_pif* program, as described in Section 9.11.

Once you have run the script until the SPE level (included) you are ready to launch the stand alone lightcurve extraction tool. *ii_light* works on the Science Window level only, meaning that you have to launch it once per Science Window. It does not change the structure of the Science Window group, so if you would like to compare the results of *ii_light* and *ii_lc_extract* it is worth to run the analysis until SPE, use *ii_light* as shown below, then go back to $REP_BASE_PROD/obs/OGID and relaunch the analysis from LCR to LCR in the standard way. The output lightcurve files of the standard software are automatically called `isgri_lcr.fits`.

Alternatively, if you have first run the standard lightcurve extraction (i.e. the LCR level) you can still run *ii_light* following the instructions below.

We give an example of the shell script that launches *ii_light* for all Science Windows of your group OGID from a particular revolution, (0051 in example below).

```
cd $REP_BASE_PROD/obs/OGID
setenv run 0051

foreach file (scw/$run*/swg_ibis.fits)
echo $file
ii_light inSwg=$file \
         num_e=4 e_min="20 40 60 100" e_max="40 60 100 200" delta_t=10 \
         outLC="$file:h/lcr.fits(ISGR-SRC.-LCR-IDX.tpl)" \
         GTIname="MERGED_ISGRI" \
         context="../../scw/$run/rev.001/idx/isgri_context_index.fits" \
         idxSwitch="../../scw/$run/rev.001/idx/isgri_pxlswtch_index.fits" \
         idxNoise="../../scw/$run/rev.001/idx/isgri_prp_noise_index.fits" \
         backDol="rebinned_back_ima.fits" \
         corrDol="rebinned_corr_ima.fits" \
         pifDOL="$file:h/isgri_pif.fits" source_selectDol="" onlydet=no \
         | tee out.log
end
```

As a result of this script you will have in each `scw/$run*` directory a file lcr.fits with 4 lightcurves (in 20 – 40, 40 – 60, 60 – 100, 100 – 200 keV energy bands) with 10 seconds binning (delta_t=10). The sources for which a lightcurve is extracted are the ones for which a PIF was created at the SPE level, i.e. specified in **SCW2_cat_for_extract** (see Section 7.2.)

9.10.3 Merge the ii_light results from different Science Windows

If you want to merge in one file all the lightcurves provided by `ii_light` for a given source, you can use the *lc_pick* tool as shown in Section 9.4.2.

ii_light is a stand alone tool and does not update the proper files with its results. Even if you have run both the lightcurve tools (in whichever order), only the standard one called within the script (LCR step,*ii_lc_extract*) will have updated the proper files with its results (i.e. the swg_ibis.fits file). Thus if you run lc_pick to collect the lightcurves, only the standard ones will be selected. If you want to collect the results of ii_light, then you have to create an index file that points to all the ii_light produced lightcurves and then give this file as input to lc_pick, as shown below:

```
cd $REP_BASE_PROD/obs/
setenv run 0051
dal_create obj_name=all_lcr.fits template=ISGR-SRC.-LCR-IDX.tpl

foreach file (OGID/scw/$run*/lcr.fits)
idx_collect index="all_lcr.fits" template="ISGR-SRC.-LCR-IDX.tpl" element="$file"
end

lc_pick source='GRS 1758-258' attach=n \
group=all_lcr.fits+1 lc=GRS1758.iilight.lc.fits emin="20"  \
emax=40 instrument="ISGRI"
```

GRS1758.iilight.lc.fits contains the merged lightcurve of GRS 1758-258 in the 20–40 keV band (that of course has to exist in the lcr.fits original files!). To display it, you can use the *lcurve* program of the FTOOLS package, as shown in Section 7.3.2.

9.11 Timing Analysis without the Deconvolution

Read this if you are interested in fast variability studies (up to milliseconds).

In this section we describe a way of doing timing analysis in a non binning way, i.e. starting from the single events. This way is suitable for very short time scales (up to milliseconds) and is less recommended for long time bins for which the binning tools *ii_light* and *ii_lc_extract* are suitable.

In the text we will use one of the Science Windows with Crab data you have downloaded to run the PICsIT analysis (e.g. 003900020020).

In general the table with the events is very big, so if you are interested in only part of the Science Window (e.g. in the case of a burst) it is better to define a user good time interval (see Section 9) and work within it.

To select the photons that come from a given source in the field of view, you need to have the corresponding PIF. PIF is automatically created during the SPE step, but can also be created with a standalone tool *ii_pif*.

Create with `og_create` observational group $REP_BASE_PROD/obs/crab/og_ibis.fits, and run analysis from COR to DEAD level, prepare the catalog, with Crab only and run *ii_pif*.

```
cd $REP_BASE_PROD/obs/crab
ibis_science_analysis ogDOL="./og_ibis.fits" startLevel=COR endLevel=DEAD

fcopy infile="$ISDC_REF_CAT[NAME=='Crab']" outfile="crab_specat.fits"

ii_pif inOG="" outOG="og_ibis.fits" inCat="../../crab_specat.fits"\
        num_band=1 E_band_min="20" E_band_max="1000"\
        mask="$REP_BASE_PROD/ic/ibis/mod/isgr_mask_mod_0003.fits"\
tungAtt="$REP_BASE_PROD/ic/ibis/mod/isgr_attn_mod_0010.fits"\
aluAtt="$REP_BASE_PROD/ic/ibis/mod/isgr_attn_mod_0011.fits"\
leadAtt="$REP_BASE_PROD/ic/ibis/mod/isgr_attn_mod_0012.fits"
```

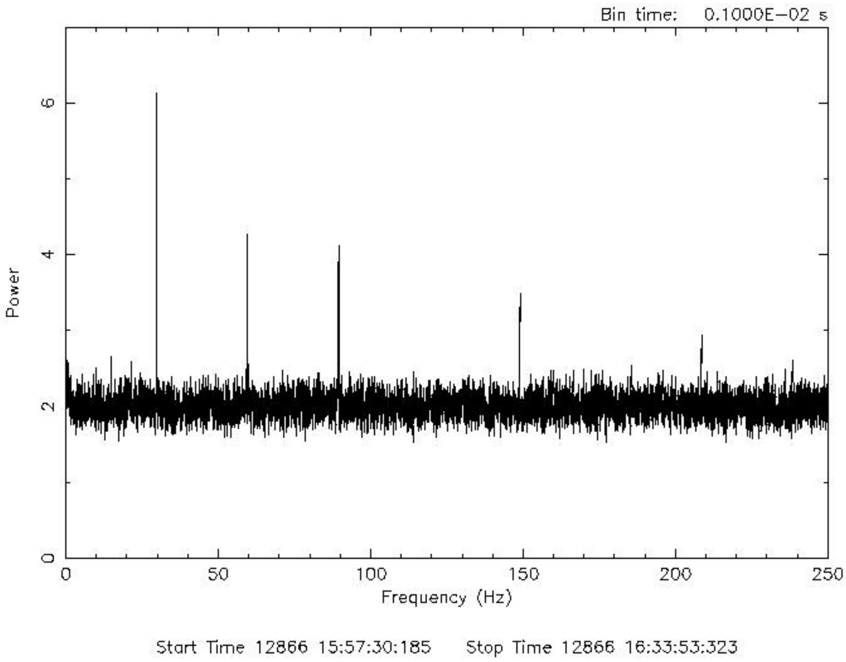


Figure 22: Crab power spectrum.

Now you are ready to create the lists of photons

```
cd $REP_BASE_PROD/obs/crab
evts_extract group="og_ibis.fits" \
events="crabevts.fits" instrument=IBIS \
sources="crab_specat.fits" gtiname="MERGED_ISGRI" \
pif=yes deadc=yes attach=no barycenter=1 timeformat=0 instmod=""
```

To increase signal-to-noise ratio select only events with PIF=1:

```
fcopy "crabevts.fits[2][PIF_1==1]" crab_pif1.fits
chmod -w crab_pif1.fits
```

Now you can produce the Crab power spectrum:

```
powspec
Ser. 1 filename +options (or @file of filenames +options)[] crab_pif1.fits
Name of the window file ('-' for default window)[] -
Newbin Time or negative rebinning[] 0.001
Number of Newbins/Interval[] INDEF
Number of Intervals/Frame[] INDEF
Rebin results? (>1 const rebin, <-1 geom. rebin, 0 none)[]  0
Name of output file[default]
Do you want to plot your results?[] yes
Enter PGPLOT device[] /XW
hardcopy crab_powerspec.ps/PS
```

As a result, the crab_powerspec.ps plot, shown in Figure 22, was produced. The 33 millisecond pulsation of the Crab is visible. For the details on *INTEGRAL* absolute timing see Walter et al. 2003 [13].

If your data have many short GTIs (e.g. in the case of telemetry saturation due to a solar flare or when PICsIT is in non standard mode) you can obtain spurious results. A typical case is finding an 8 sec period in your data due to the fact that the telemetry restart is synchronized with an 8 sec frame! When possible, compare your results with ii_light that is immune to this problem and can reach about 0.1 sec binning.

9.12 Phase Resolved Analysis

Read this if you want to perform phase resolved analysis

There are no special tools in OSA 10 for the phase resolved analysis. Below we give some hints on how to apply the existing tools for your needs in different cases. In general three cases are possible:

1) The time scale you are interested in is bigger than several hours.

In this case you should just select the Science Windows matching your time selection criteria. For this you can use e.g. the keywords TSTART and TSTOP in the header of swg.fits.

2) The time scale you are interested in is in the range of seconds to hours.

In this case the GTI usage is recommended, see Section 9.2. Note that if the number of USER GTIs exceed 1000, then the program could become unreasonably slow.

3) You are interested in very short (milliseconds) time scale. Currently this case requires lots of manual work. The idea is to use `SCW1_ISGRI_event_select` parameter to define your CFITSIO criteria for the phase selection. Unfortunately it is not possible to apply this criterion directly to the TIME column, and you will have to modify the DELTA_TIME column of the third (ISGR-EVTS-ALL) extension of the `isgri_events.fits`. This file is usually write protected, so in general you will have to copy all the data related to the science window of interest to a new place, and change links.

In the example below we are working with science window 005100570010, and are interested in $\Delta T = 20$ milliseconds (0.02 s) time bins with zero phase at T0=1170.36 (in IJD). The general selection criterion can be written as:

$Ph1 < modulus((T - T0) * 24 * 3600, \Delta T) * 100/\Delta T) < Ph2.$

Here T and T0 are expressed in days, ΔT in seconds, and phase period is split into 100 intervals, *i.e.,* the phase written to column DELTA_TIME changes from 0 to 99. In our example it will look like:

```
cd $REP_BASE_PROD
rm -r scw
mkdir scw
mkdir scw/0051
cd scw/0051
cp  directory_of_local_archive__/scw/0051/005100570010.001 .
ln -s  directory_of_local_archive__/scw/0051/rev.001   .
chmod +w   005100570010.001
cd  005100570010.001
chmod +w  isgri_events.fits.gz
fcalc isgri_events.fits.gz\[3] isgri_events_1.fits clname="DELTA_TIME"\
      expr="(((TIME-1170.36)*24*3600)%0.02)*100/0.02"
gzip isgri_events_1.fits
mv   isgri_events_1.fits.gz  isgri_events.fits.gz
chmod -w  isgri_events.fits.gz
cd ../
chmod -w  005100570010.001
```

```
cd  $REP_BASE_PROD
og_create idxSwg=phase.lst ogid=phase  baseDir="./" instrument=IBIS
cd obs/phase
ibis_science_analysis SCW1_ISGRI_event_select='DELTA_TIME <50'
```

Note that in this case you create a dead time about which the software is not aware of. Thus you will have to correct the resulted flux by the factor (original exposure)/(real exposure). In the case above we have selected events with phase less than 50, i.e. half of the time was rejected, and the resulted flux should be multiplied by 2.

10 PICsIT data analysis

Unfortunately PICsIT sensitivity is not high enough to create a good image for a single Science Window, thus you should expect good results only if *INTEGRAL* is in staring mode (and the source is really bright!). In the example below we use Crab observations done in staring mode.

Download and install, as described in Section 6.1, the following Science Windows, whose DOLs are assumed to be written to the file **picsit.lst**:

```
scw/0039/003900020020.001/swg.fits
scw/0039/003900020030.001/swg.fits
scw/0039/003900020040.001/swg.fits
scw/0039/003900020050.001/swg.fits
scw/0039/003900020060.001/swg.fits
```

Create the working directory `picsit_ima` with the *og_create* program:

```
cd ibis_data_rep
setenv REP_BASE_PROD $PWD
og_create idxSwg=picsit.lst ogid=picsit_ima  baseDir="./" instrument=IBIS
cp -r obs/picsit_ima obs/picsit_spe
```

10.1 PICsIT Image Reconstruction

To start the analysis, go to the working directory **$REP_BASE_PROD/obs/picsit_ima** and call the *ibis_science_analysis* script:

```
cd obs/picsit_ima
ibis_science_analysis
```

After a few seconds, the main page of the IBIS Graphical User Interface (GUI) appears, as shown in Figure 23. Press "Reset"-button to be sure that you have the default ISDC_ENV parameters.

For PICsIT there were no major changes since the creation of revision 2 data, so you can start directly from the BIN_I level (startLevel=BIN_I). The end level for PICsIT image creation should be the default `endLevel=IMA2`. Disable ISGRI analysis by checking SWITCH_disableIsgri button, and enable PICsIT by unchecking SWITCH_disablePICsIT button.

PICsIT operates in an energy range (0.175-10 MeV) where background subtraction plays a very important role. The OSA for PICsIT provides a default set of maps for the background subtraction in certain energy bands, selected to optimize the instruments performance. By default, the software will automatically take subtraction with one set of maps (with about 1.7 Ms of integration time). The default energy bands are (in

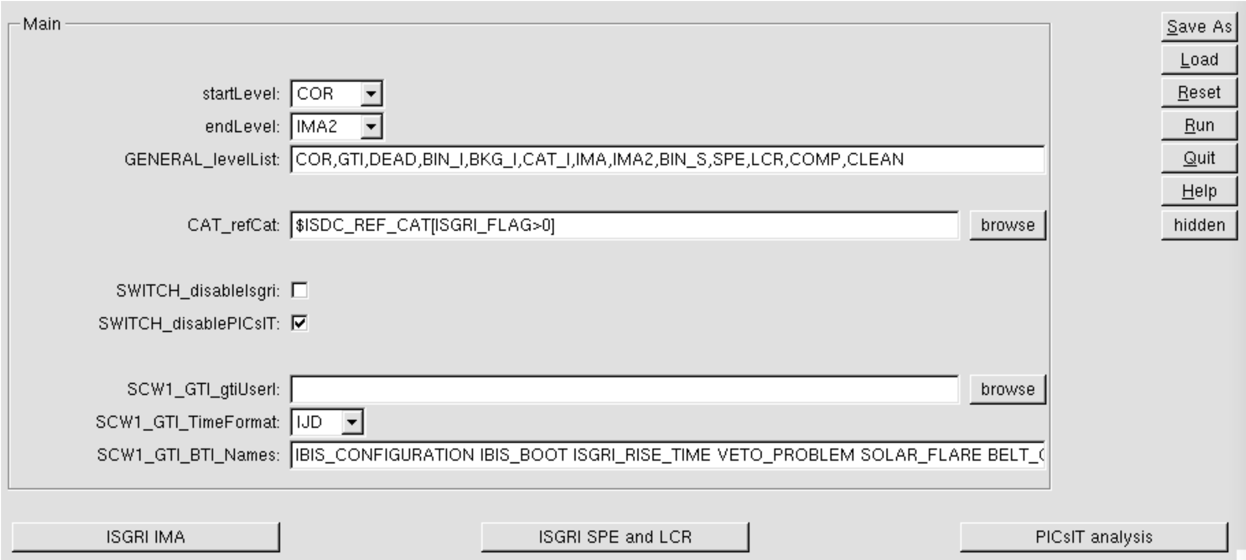


Figure 23: Main page of the IBIS GUI

keV) for single events: 203 – 252, 252 – 336, 336 – 448, 448 – 672, 672 – 1036, 1036 – 1848, 1848 – 3584, 3584 – 6720. For multiple events: 336 – 448, 448 – 672, 672 – 1036, 1036 – 1848, 1848 – 3584, 3584 – 6720, 6720 – 9072, 9072 – 13440.

This works for standard mode (mode 41) [17] only.

To select parameters specific for PICsIT analysis press the `PICsIT` button. The PICsIT GUI page is shown in Figure 24.

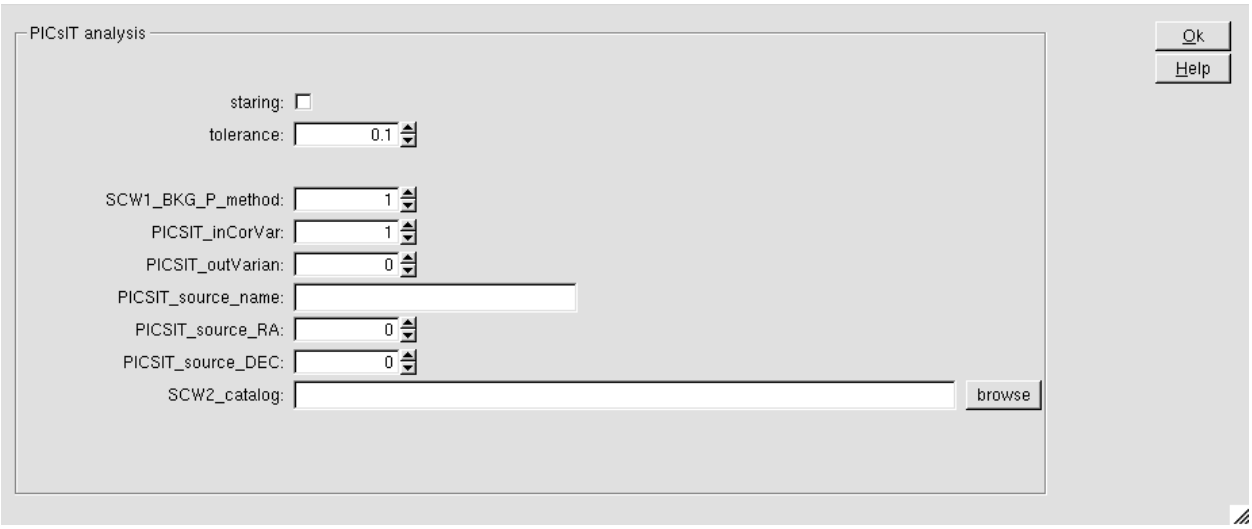


Figure 24: PICsIT page of the IBIS GUI

Depending on the observation mode (staring or dithering), there are two ways to analyze the PICsIT data. The first, by using the parameter `staring=yes`, can be used only for staring observation, where the difference between all the pointings is less than the value defined by the `tolerance` parameter(for PICsIT use `tolerance=0.1`). In this case, the corrected shadowgrams are integrated before the deconvolution. If `staring=no`, then deconvolution will be performed for each science window individually.

[17] The IBIS data mode can be checked in the archive by selecting IBIS mode column on the page with additional parameters

The science windows we use in the current example were observed in staring mode, so check the staring button.

`SCW1_BKG_P_method` defines the way to normalize the background maps to the observed shadowgram before performing the background subtraction. 0 means scaling with exposure, and 1 means scaling with the average counts value over the whole detector. Both generally provide the same results, though the second method, which serves as default, is sometimes a bit better.

`PICSIT_inCorVar` defines whether you would like the software to partially correct the variance shadowgram for systematic effects due to the background (value 1), or not, in order to estimate by yourself the degree of systematics present in the deconvolved maps (value 0).

`PICSIT_outVarian` defines whether you would like to have the variance maps in output (value 1), or not (value 0). Note that the *varmosaic* tool of HEASOFT needs variance maps to work.

Return to the main GUI window by clicking on OK. Without changing other default parameters, you are now ready to run the analysis; do so by clicking on Run.

10.1.1 Results of PICsIT image analysis

In case of a staring the output images are in file

```
./picsit_ima.fits
```

For each energy range and each type of event (single, multiple), 2 types of images, INTENSITY and SIGNIFICANCE, are produced. In case `PICSIT_inCorVar`=1, a variance map is also produced.

The list of found sources is in the file

```
./picsit_sky_res.fits
```

In our example the Crab is the only real source that was found.

In Figure 25 you see the resulting image in the $252 - 336$ keV energy range. The Crab is clearly seen in the center. The figure was produced with the help of ds9:

```
cat2ds9 isgri_catalog.fits cat.reg symbol=circle color=red
ds9 picsit_ima.fits\[5] -region cat.reg \
                           -cmap cool -scale linear -scale limits 0 7 -zoom 8
```

In the case of observations with dithering pattern (`staring=no`), the pipeline performs the shadowgram deconvolution in every Science Window and the output files are under:

```
./scw/RRRRPPPPSSSF.001/picsit_ima.fits
./scw/RRRRPPPPSSSF.001/picsit_sky_res.fits
```

and the integration of all these images (mosaic) is in:

```
./picsit_ima.fits
```

10.2 PICsIT spectral extraction from the mosaic image

It is also possible produce the source spectrum from the mosaic image.

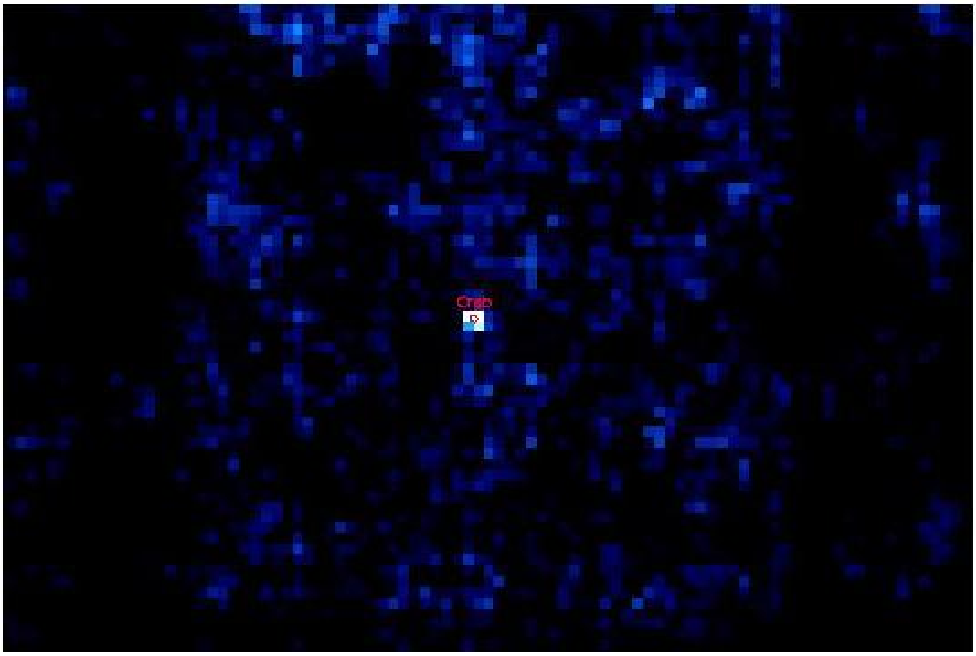


Figure 25: Crab significance image in the $252 - 336$ keV energy band as seen by PICsIT.

At http://www.isdc.unige.ch/integral/osa/scripts, you find an example of such a of perl script *spextract_pics.pl*. To use it create a list of files you want to use for spectral extraction. In the case of staring, file `picsit.lst` will contain a single line:

```
picsit_ima.fits
```

Now you are ready to extract the spectrum. To do this you can either fill the values manually from the `picsit_sky_res` file, or extract the fluxes from the **brightest** pixel around the catalog position with the help of the *spextract_pics.pl*:

```
cd $REP_BASE_PROD/obs/picsit_ima
perl spextract_pics.pl -i picsit.lst -r 83.605 -d 21.95\
-n 8  -o crabspe_pics.fits  -m ../../ic/ibis/rsp/pics_srmf_grp_0005.fits\
-a ../../ic/ibis/rsp/pics_sarf_rsp_0003.fits
```

(In this command, switches -i and -o define the input and output files, -r and -d define the RA and DEC of the desired pixel, -n defines the number of energy ranges, -m and -a define the locations of rmf and arf matrices).

The result is shown in Figure 26.

10.3 PICsIT Timing Analysis

It is possible to produce a lightcurve of the PICsIT detector from the spectral-timing data. To build it, it is enough to run the analysis from the COR to DEAD level and then from LCR to LCR (of course, it is not a problem if you have already done the analysis up to the IMA2 level). Do not forget to disable ISGRI and enable PICsIT for the analysis. The results are in the following FITS files:

```
./scw/RRRRPPPPSSSF.001/picsit_lcr.fits
```

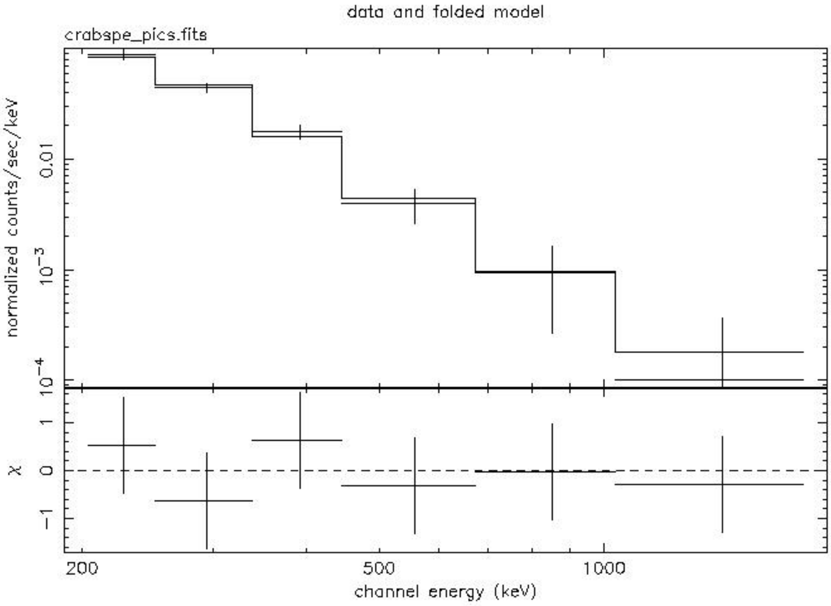


Figure 26: PICsIT Crab spectrum extracted from the mosaic.

These files can be analyzed with standard `xronos` tools.

11 Known Limitations

Please note, that in the time between the releases this list can change. For the most updated version see

http://www.isdc.unige.ch/integral/osa/current/issues_osa

11.1 ISGRI

1. Systematic uncertainties of 1% should be added to ISGRI counts, fluxes.
2. In the mosaic build with the option `spread=1` the source flux is slightly reduced (∼10 %) compared to the weighted average of the fluxes measured in the Science Window.
3. The maximum number of sources handled by ii_spectra_extract is 200 but it is strongly recommended to only fit spectra of the sources that are effectively active (visible, detectable) during the Science Window.
4. With OSA10, new calibration files have been produced including a correction for the variation of gain across the entire mission, as observed in previous OSA versions. However, on single revolution time scale, a drift in counts is still observed. For the latest part of the mission, spectra extracted at the beginning and end of a same revolution can therefore show an artificial difference in counts. The secular drift observed in all bands over the mission life-time is known and due to the evolution of gain: this effect is accounted for by the set of ARFs available in the IC tree.
5. The position of the low-energy threshold is increasing with time (see Sect. 12.4.1). A safe lower limit for the response is 18 keV until revolution 848.
   Between revolutions 848-1090, we recommend to ignore data below 20 keV.
   From revolutions 1090 on, we recommend the user to ignore data below 22 keV.
6. A problem on-board IBIS causes event times to be shifted by 2 seconds under some circumstances (this is rare). The software tries to correct the data. The keyword TIMECORR found in the event files (*-*-ALL or *-*PRP extensions), indicates whether the correction was done. If you are doing an accurate timing analysis and your data contains TIMECORR>0 please take great care: If TIMECORR=1 or 2, the applied correction should be OK. If TIMECORR=3 you should better not use these data. If TIMECORR=4 contact ISDC.
7. The lightcurve extraction (ii_lc_extract) is performed by building shadowgrams for each time and energy bin. It potentially takes a large amount of CPU time and there is a minimum usable time bin. The time bin must be such that the total number of maps in the file isgr-corr-shad does not exceed 2 GB worth of disk space. The product of the number of time bins in a science window, and the number of energy bands must be less than about 9942.
8. ii_pif will crash if the input catalog inCat contains more than 500 sources.
9. At large off-axis angles the IBIS response is not well known and strongly energy dependent. Therefore, the user should be careful when analyzing observations performed at large off-axis angles, above ∼12 degrees, since systematic flux variations might be introduced. The systematic flux variations are energy dependent, and therefore the user should be careful both with photometric and spectral analysis of sources at large off- axis angles.

11.2 PICsIT

1. The spectra extraction with the PIF method is not reliable for the moment (executable *ip_spectra_extraction*). The user should extract the spectra from images (count rates from intensity maps and errors from significance maps) and then convolve them with the RMF/ARF.

Part III

Data Analysis in Details

12 Science Analysis

In the Cookbook you have seen that in order to run the Scientific Analysis you should just launch the main script *ibis_science_analysis* with a desired set of parameters. As discussed in the Overview (Section 5) the main script consists of smaller scripts, which in turn unify executables with similar tasks for different types of events (see Figure 27). In this chapter we describe these small scripts in more detail in order to explain how the main script works and which parameters you have to enter for a proper analysis.

While describing the executables we describe those parameters which were included into the main script. You find in the Appendix the detailed description of the results produced at each step. In the Appendix you also find the description of raw and prepared data with which you start the analysis.

12.1 *ibis_correction*

This script produces corrected Data Structures for all types of the events. It combines the following executables:

- *ibis_isgr_evts_tag*
- *ibis_isgr_energy*
- *ip_ev_correction*

Using the information about the noisy pixels (see Section 3.2.3) the script flags all the noisy events. Then the script performs the energy correction of all the events for which correction is possible (no correction can be done for the spectral-timing mode). The output Data Structures contain the list of the photon energy in keV (see Appendix C.1).

12.1.1 *ibis_isgr_evts_tag*

ibis_isgr_evts_tag tags, as noisy, all photons from a pixel if this pixel had a switch off during the ScW. It also looks at the distribution of the time between events in each given pixel. If this distribution is abnormal, all photons from this pixel are flagged as noisy (within this ScW). Noisy events are ignored in the subsequent analysis.

Table 4: *ibis_isgr_evts_tag* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
SCW1_ICOR_idxSwitch	idxSwitch	string	DOL of the index of pixels switch list. default: “ ”
SCW1_ICOR_probShot	probShot	real	Probability of shot time decay. default: 0.0001

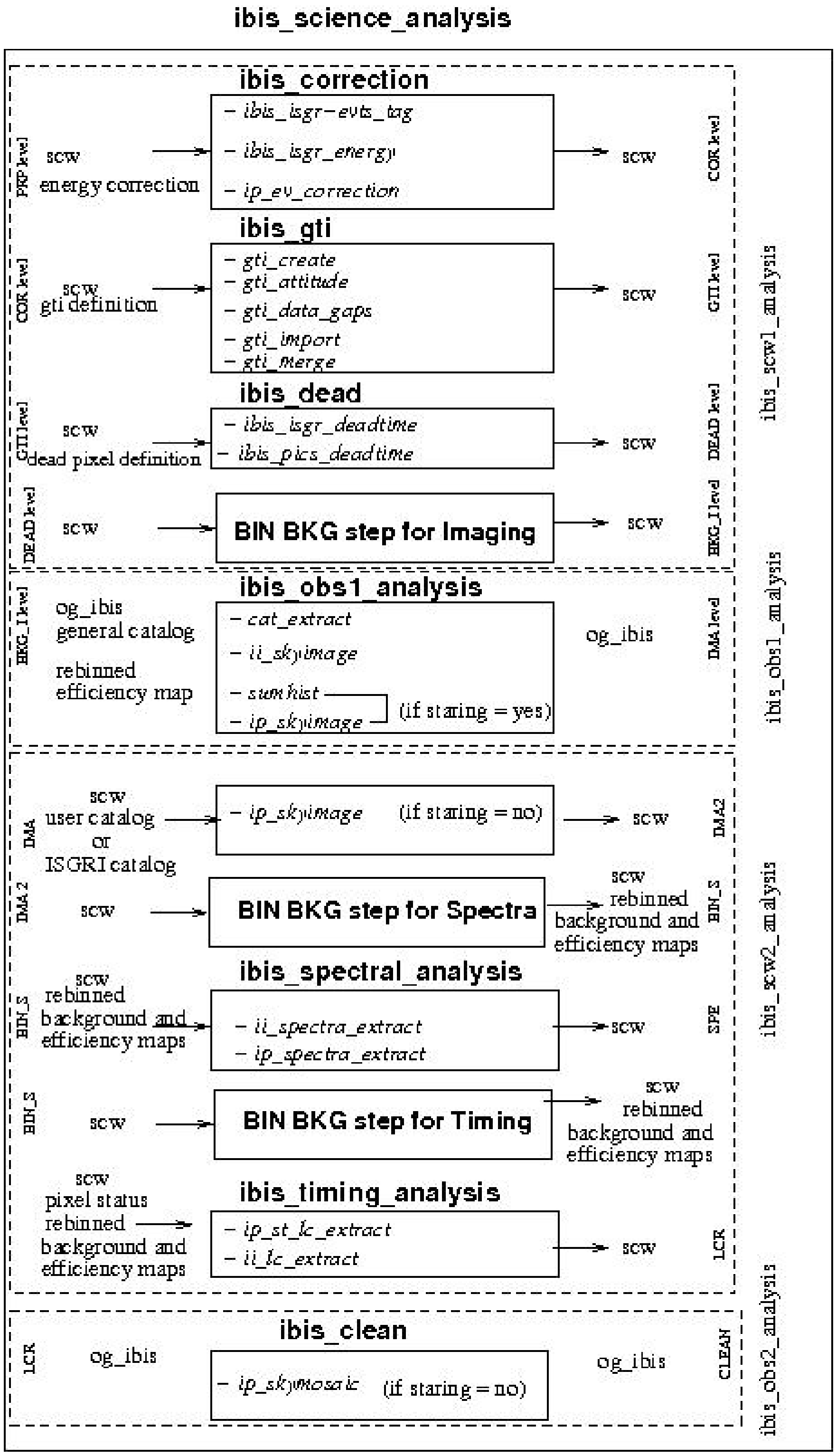


Figure 27: Composition of the main script *ibis_science_analysis*. For further descriptions of the `BIN BKG` steps for the `DEAD`, `IMA` and `BIN_S` levels, see Figures 28, 29 and 30 respectively.

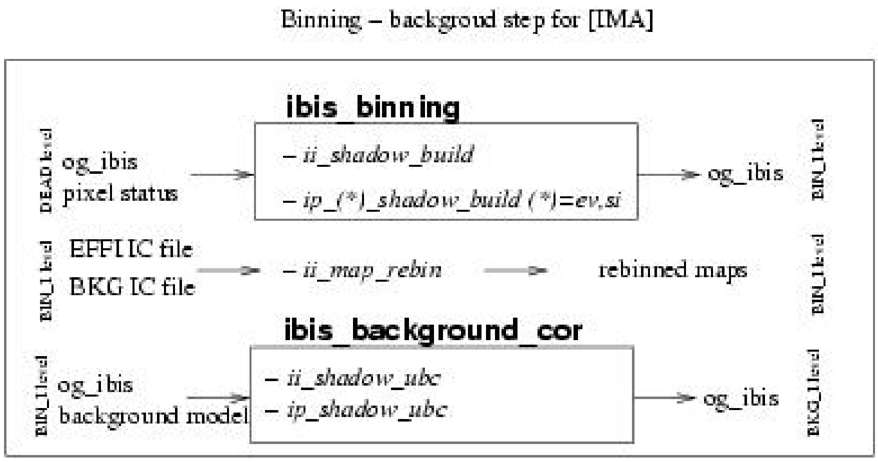


Figure 28: Overview of the binning - background step for Imaging.

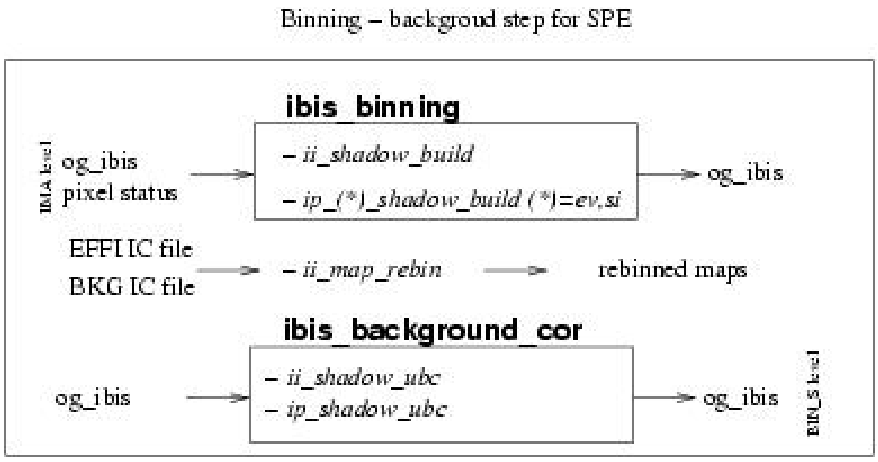


Figure 29: Overview of the binning - background step for Spectra.

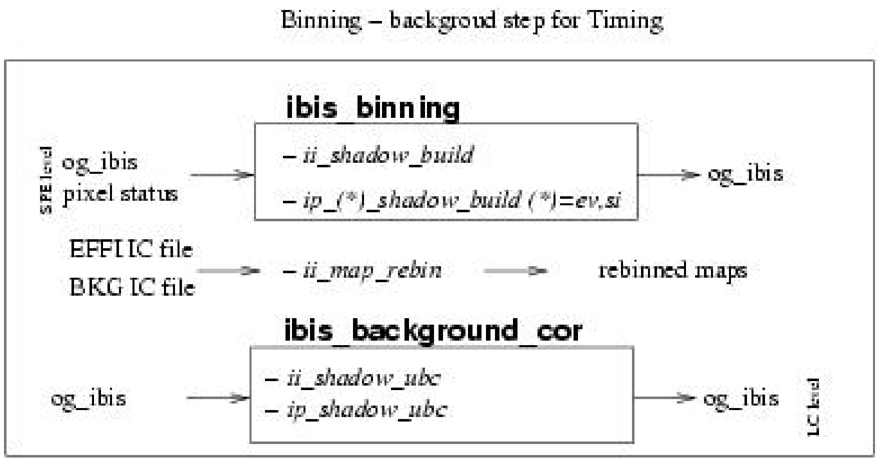


Figure 30: Overview of the binning - background step for Lightcurves.

12.1.2 *ibis_isgr_energy*

Due the charge loss in the CdTe crystal, for a given energy deposit, events which have interacted at different depths in the crystal get different measured amplitudes (PHA). The rise-time of the signal induced by these events is also different, and a clear quasi-linear correlation between the charge loss and the rise-time variation is observed (a deeper interaction induces a larger charge loss, giving the longer rise-time). It is then possible to correct this charge loss effect, by taking into account the rise-time information of the signal.

The executable *ibis_isgr_energy* performs a rise-time correction for each raw ISGRI event, using the ISGRI rise-time correction table (Section B.2). The corrected energy is given in keV (`ISGRI_ENERGY`), and the corrected rise-time (`ISGRI_PI`) is the row number used in the IC file. The correction is done by rescaling the measured spectra in accordance with the observed rise-time `rt`, so that the corrected event corresponds to the deposited energy.

Also the temperature and bias corrections are done at this step.

The ISGRI spectral gain has been observed to decrease with time. In OSA9 the description of the gain drift was based on IREM counters integrated over time, to take into account the solar flares. However, this correction proved to be not stable along the whole mission. In OSA10, in order to stabilize the instrument response across the whole mission, a refined dependence of the gains-offsets correction with the (measured) temperature of the different MDU has been introduced. A similar correction was performed also in OSA versions later than OSA 7 assuming, however, a constant ΔT between the different modules, which is not a correct assumption along the mission.

The new OSA10 calibration results in an increased stability of the W and ^{22}Na line positions along the mission, and in a minor dispersion of Crab and background spectra along the entire mission compared to previous OSA versions. More details on the new energy calibration can be found at [14]

Table 5: *ibis_isgr_energy* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
SCW1_ICOR_GODOL	GODOL	string	DOL of the Gain-Offset correction table default: “ ”
SCW1_ICOR_riseDOL	riseDOL	string	DOL of the rise-time correction table default: “ ”
SCW1_ICOR_supGDOL	supGDOL	string	DOL of gain coefficients for 2nd method default: ‘‘ ’’
SCW1_ICOR_supODOL	supODOL	string	DOL of offset coefficients for 2nd method default: ‘‘ ’’

12.1.3 *ip_ev_correction*

The executable *ip_ev_correction* performs energy correction of the events received by PICsIT in photon-by-photon mode. Single events are corrected for each pixel separately with the use of the pixel-dependent `gain` and `offset` factors, and the pixel-independent channel-to-keV conversion factors `avgain` and `avoffset`, see Section B.2 for more details.

$$\texttt{energy[keV]} = \texttt{avgain} \cdot \texttt{gain} \cdot \mathrm{PICSIT\_PHA} + \texttt{avoffset} \cdot \texttt{offset}$$

Multiple events are corrected for gain and offset on-board, and thus they are simply transferred from channels to keV:

$$\texttt{energy[keV]} = \texttt{avgain} \cdot \texttt{PICSIT\_PHA}$$

Table 6: *ip_ev_correction* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
SCW1_PCOR_enerDOL	enerDOL	string	DOL of the energy correction table (PICsIT). default: " "

12.2 *ibis_gti*

This script builds Good Time Interval (GTI) information from housekeeping data, information about the satellite stability and data gaps. It calls the following executables to obtain the GTIs:

- *gti_create*
- *gti_attitude*
- *gti_data_gaps*
- *gti_import*
- *gti_merge*

For the definition of GTI see Introduction to the INTEGRAL Data Analysis [1]

12.2.1 *gti_create*

This program generates all GTIs for one instrument that depend on HK and other parameters and are defined by a limit in a limit table, see details in Section B.3. In ISGRI case it creates **VETO** GTIs during which VETO was switched on, and **ISGRI_MCE7X** (where X varies from 0 to 7) GTIs during which the Xth module was switched on. **VETO** GTIs are taken into account by *gti_merge* executable (see Section 12.2.5), and **ISGRI_MCE7X** GTIs are taken into account later during BIN_I, BIN_S and BIN_T levels.

Table 7: *gti_create* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
SCW1_GTI_LimitTable	LimitTable	string	The DOL of the GTI limit table. default: " "

12.2.2 *gti_attitude*

A GTI is defined for each period of time where the pointing stability is better than the accepted tolerance (parameter AttStability). This GTI is named "ATTITUDE". For slews this GTI is always set to be good independently of any input data. If the necessary data are not available the GTI is set to be "not good" for the whole Science Window.

Table 8: *gti_attitude* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
SCW1_GTI_attTolerance_X	AttStability	real	Accepted attitude stability tolerance of X (optical) axis to generate a GTI [arc min] possible values: `0 – 10800` default: `0.5`

SCW1_GTI_attTolerance_Z	AttStability_Z	real	possible values: `0 – 10800` default: `3.0`

12.2.3 *gti_data_gaps*

This program generates GTIs for ISGRI and PICsIT. The GTIs depend on the presence of the science data. A time is defined as bad if a science packet of the instrument is missing. The expected science packets depend on the actual mode of the instrument. This is taken into account while the GTI is being created. The program does not distinguish between a pointing, a slew and an engineering window. The time is also set to be "bad" if there are scientific data that do not agree with the reported mode of the instrument.

Time is also defined bad if the telemetry indicates that there was an interruption of the on-board processing due to a telemetry saturation. In normal operation this case is rather seldom.

The final GTI is written into the IBIS index group and has the name "ISGRI_DATA_GAPS" and "PICSIT_DATA_GAPS" for ISGRI and PICsIT correspondingly.

12.2.4 *gti_import*

The *gti_import* reads the user GTI table and converts it into a table in ISDC format. The user GTI can be defined either in units of OBT, IJD, or UTC. The output is always in OBT. The user table can define either bad or good time intervals. The output time intervals are always good ones. See more details in the Introduction to the *INTEGRAL* Data Analysis [1].

Table 9: The *gti_import* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
SCW1_GTI_gtiUser	InGti	string	DOL of the user GTI table " "=there is no user GTI. default: " "
SCW1_GTI_TimeFormat	TimeFormat	string	Time format in which the user GTI is given. possible values: "`IJD`", "`UTC`", "`OBT`" default: "`UTC`"
SCW1_GTI_Accuracy	Accuracy	string	Used accuracy for OBT to IJD conversion and vice versa. possible values: "`any`", "`inaccurate`", "`accurate`" default: "`any`"

12.2.5 *gti_merge*

This program merges input GTIs to form a new GTI. It is an AND operation: a time in the resulting GTI is defined to be "good" if the time is "good" in every input GTI. The names of the GTIs and the instrument to which the GTI belongs have to be defined as program parameters.

Table 10: *gti_merge* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
SCW1_GTI_ISGRI	MergedName	string	GTIs to be merged for ISGRI. default: `''VETO ATTITUDE ISGRI_DATA_GAPS''`

SCW1_GTI_PICsIT	MergedName	string	GTIs to be merged for PICsIT. default: `''VETO ATTITUDE P_SGLE_DATA_GAPS P_MULE_DATA_GAPS''`
SCW1_GTI_SCI	SC_Names	string	Names of spacecraft GTIs to be merged for ISGRI default: `'' ''`
SCW1_GTI_SCP	SC_Names	string	Names of spacecraft GTIs to be merged for PICsIT default: `'' ''`
SCW1_GTI_SCP	BTI_Dol	string	DOL of a bad Time interval table (GNRL-INTL-BTI) default: `'' ''`

12.3 *ibis_dead*

This script combine two executables calculating the dead time for ISGRI, PICsIT and Compton cases.

- *ibis_isgr_deadtime*
- *ibis_pics_deadtime*

The detector efficiency is divided into the real efficiency and the dead time D which is due to the delay following a photon detection during which another event can not be recorded.

The observed count rate $C_{\rm o}$ is connected with the real count rate C as

$$C_{\rm o} = C(1 - D)$$

The presence of the BGO shielding (see Section 3.2.5), calibration source, and the Compton coding induces an efficiency loss due to good events which by chance fall in their respective time windows.

For IBIS the term "efficiency" is reserved for the real physical detection efficiency. The effects induced by Veto, calibration source, and Compton tagging are considered as additional dead times to be added to instrumental dead time.

Numerically, dead time is calculated as the product of the count rate and the time window (e.g. the dead time due to the time of the photon detection is equal to the product of the module count rate and the coding time. The dead time due to the veto is equal to the product of the veto count rate and the veto time window, etc.) The resulting dead time is taken as a sum of all the dead times connected with different effects. Simple summing of dead times can be done since the estimations show that the coincidence probability between different types of events is low. The values of the count rates is taken from the HK Data Structures **IBIS-DPE.-HRW** and **IBIS-DPE.-CNV** and the corresponding time windows are either measured (instrumental) or read into HK or IC (all others: VETO, Compton).

12.3.1 *ibis_isgr_deadtime*

The executable *ibis_isgr_deadtime* calculates dead times resulting from count rates in ISGRI modules, taking into account the instrument configuration.

ISGRI dead time is due to the combined effect of:

- Instrument count rate,
- Veto count rate (from the lateral, or lateral+bottom (according to IC file) shielding),
- Calibration count rate,
- Compton coincidences, if Compton mode is on.

Note that the problem of the VETO swapping on-board is corrected almost every time through IC file: **IBIS-VETO-MOD**. VETO also sometimes has a wrong level; this will be corrected in the future. The 3 cases when it happened are in the IC file **GNRL-INTL-BTI** with the BTI_TYPE: **ISGRI_VETO_LEVEL**.

Table 11: *ibis_isgr_deadtime* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
SCW1_veto_mod	icDOL	string	DOL of the IC file for VETO model and width of Compton window default “ ”

12.3.2 *ibis_pics_deadtime*

The executable *ibis_pics_deadtime* extracts from the housekeeping data the PICsIT intrinsic dead time of and dead time induced by the fortuitous coincidence with VETO and calibration system.

Contrary to the ISGRI case, the dead time is measured by the on-board electronics and we have only to decipher it from the telemetry data.

12.4 *ibis_binning*

The main function of *ibis_binning* is to split the data into energy and time bins. The number of the time and energy bins depend on the purpose of the analysis. The user can choose if the binning is done for imaging, spectral extraction, or light curve production. Typically just one bin in time and few bins in energy are chosen for the image production; one time bin and many energy bins for spectral extraction; many time bins and just few energy bins for timing analysis.

This step is repeated for each Science Window in the observational group. The shadowgram is created for each energy interval. Script *ibis_binning* combines the following executables:

- *ii_shadow_build*
- *ip_ev_shadow_build*
- *ip_si_shadow_build*

12.4.1 *ii_shadow_build*

The executable *ii_shadow_build* creates shadowgrams of ISGRI events in the given time (Tbin) and energy (Ebin) ranges. For every shadowgram a corresponding efficiency shadowgram is created. For pixels active during the revolution (switch status ON), efficiency falls into two components and is null for the others: the first factor takes into account the dead time D of the corresponding module MDU; the second factor reflects the efficiency energy dependence, LT, for the lower energy bins due to the low threshold operation limit, such that: Eff[Tbin,Ebin,y,z]= (1-D[Tbin,mdu]) * LT[Ebin,y,z]. It is this value that is given in the shadowgram efficiency maps. The low threshold (LT) correction is performed using the same energy correction of the COR step and an erf function with time-dependent width which reproduces the degradation in energy resolution of the instrument. The position of the low threshold is evolving with time as can be seen in the Fig. 31, with the ISGRI efficiency being lower than 50% below this energy. As an example, for revolutions later than Rev.1000, energies below $\sim$22 keV should be ignored.

This executable also applies two different noisy pixel detection techniques. The first one is time-based; it discriminates every pixel for which the scientific SELECT_FLAG equals one (even only once over all the

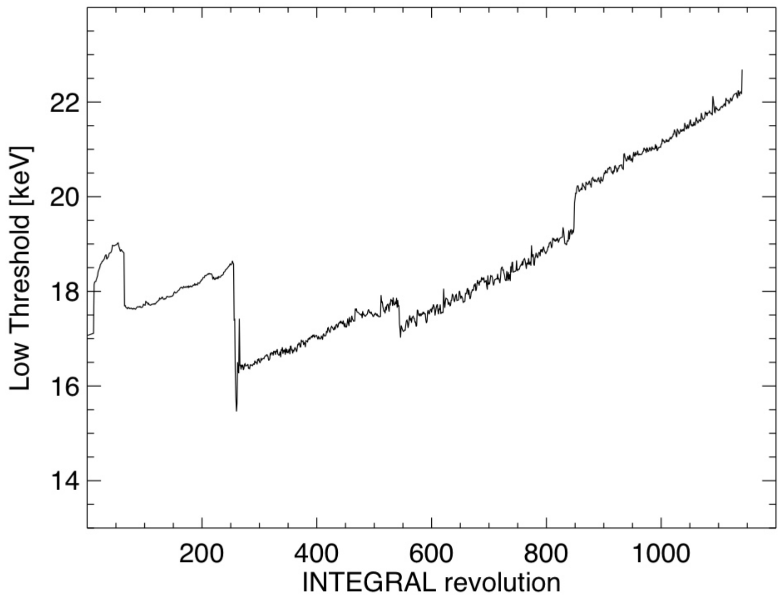


Figure 31: ISGRI low threshold position as function of INTEGRAL revolution number.

Science Window events). Note: this flag was filled by the executable *ibis_isgr_evts_flag*. The second method is spectral-based; it discriminates pixels by estimating their spectral deviation from the average ISGRI spectrum. This spectral method can be switched ON/OFF by parameter NoisyDetFlag.

Table 12: *ii_shadow_build* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
IBIS_II_ChanNum	isgri_e_num	integer	Number of energy intervals for ISGRI. possible values: `1 – 10` default: `7`
IBIS_II_E_band_min	isgri_e_min	string	List of lower energy boundaries (keV). default: "`15 40 100 200 400 700 1200`"
IBIS_II_E_band_max	isgri_e_max	string	List of upper energy boundaries (keV). default: "`40 100 200 400 700 1200 2500`"
IBIS_min_rise	isgri_min_rise	integer	Minimum rise-time. default: `16`
IBIS_max_rise	isgri_max_rise	integer	Maximum rise-time. default: `116`
IBIS_NoisyDetMethod	NoisyDetFlag	integer	Defines the way to deal with noisy pixels possible values: `0` Time-based noisy detection; `1` Time-based + Spectral-based method. default: `1`

12.4.2 *ip_ev_shadow_build*

ip_ev_shadow_build takes as an input PICsIT data received in photon-by-photon mode. For each given energy and time range, intensity and efficiency shadowgrams are produced. Efficiency is defined as:

$$\mathrm{Eff} = (1 - D) * T_{\mathrm{ON}}/T$$

where T_{ON}/T is given by the GTI and the total observation length.

Table 13: *ip_ev_shadow_build* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description

IBIS_IP_E_band_max_m	picsit_e_max_m	string	List of higher energy boundaries for PICsIT (multiple events) default: "`600 1000 13500`"
IBIS_IP_E_band_min_s	picsit_t_bin_s	string	Time bins [for the lightcurves] default: "`1`"
SCW1_BIN_P_inDead	inDead	string	DOL of the dead time Data Structure. default " "
SCW1_GTI_PICsIT	gti_name	string	Name of GTI to use. default: `''VETO ATTITUDE P_SGLE_DATA_GAPS P_MULE_DATA_GAPS ''`
SCW1_BIN_P_PicsCxt	PicsCxt	string	DOL of the PICsIT Context Tables. default: `'' ''`

12.4.3 *ip_si_shadow_build*

ip_si_shadow_build takes as an input PICsIT data received in standard mode. For each given energy and time range, intensity and efficiency shadowgrams are produced. Efficiency is defined as:

$$\mathrm{Eff} = (1 - D) * T_{\mathrm{ON}}/T$$

Table 14: *ip_si_shadow_build* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
IBIS_IPS_ChanNum	picsit_e_bin	integer	Number of Energy bands (`-1` means to use the default ones.) default: `8`
IBIS_IPS_E_band_min_s	picsit_e_min_s	string	List of lower energy boundary (single events). default: `203 252 336 448 672 1036 1848 3584`
IBIS_IPS_E_band_max_s	picsit_e_max_s	string	List of higher energy boundary (single events). default: `252 336 448 672 1036 1848 3584 6720`
IBIS_IPS_E_band_min_m	picsit_e_min_m	string	List of lower energy boundary (multiple) default: `336 448 672 1036 1848 3584 6720 9072`
IBIS_IPS_E_band_max_m	picsit_e_max_m	string	List of higher energy boundary (multiple) default:`448 672 1036 1848 3584 6720 9072 13440`
SCW1_BIN_P_inDead	inDead	string	DOL of the dead time Data Structure. default " "
SCW1_BIN_P_PicsCxt	PicsCxt	string	DOL of the PICsIT Context Tables. default: `'' ''`
SCW1_BIN_P_HepiLut	HepiLut	string	DOL of the hepi look-up-table (lut) default: `'' ''`
IBIS_IPS_corrPDH	corrPDH	integer	Packets threshold for partially downloaded histograms. default:`0`
IBIS_P_convFact	convFact	real	Conversion factor channel-to-energy. default: `7.1`

12.5 *ii_map_rebin*

ii_map_rebin reads input background and/or Off-axis correction maps and rebins them into new energy

bands. Background maps are summed with coefficients equal to the part of each input map that participates in the sum. Off-axis maps are also weighted by a power-law spectrum (with the default slope -2) and ARF (if present). After rebinning, off-axis maps are normalized to 1. Output energy bands are defined from the index of detector shadowgrams (ISGR-DETE-SHD) of a given type.

Table 15: *ii_map_rebin* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
corrDol	inpCorrDol	string	DOL of the isgri off-axis corrections default " "
rebinned_corrDol_ima	rebCorrDol	string	DOL of the rebinned isgri off-axis corrections for imaging default " "
rebinned_corrDol_spe	rebCorrDol	string	DOL of the rebinned isgri off-axis corrections for spectrum default " "
rebinned_corrDol_lcr	rebCorrDol	string	DOL of the rebinned isgri off-axis corrections for lc default " "
rebinned_backDol_ima	rebBkgDol	string	DOL of the rebinned isgri background corrections for imaging default " "
rebinned_backDol_spe	rebBkgDol	string	DOL of the rebinned isgri background corrections for spectrum default " "
rebinned_backDol_lcr	rebBkgDol	string	DOL of the rebinned isgri background corrections for lc default " "
SCW1_BKG_I_isgrBkgDol	inpBkgDol	string	DOL of the isgri background model default " "

12.6 *ibis_background_cor*

This script combines executables which fill the dead zones of the detectors in accordance with the chosen method, creating the extended intensity map I_{ex}. Then with the help of the IC background, B, and uniformity, U, maps (See Section B.4) a corrected intensity map I_{cor} is produced:

$$I_{cor} = \frac{I_{ex} - B}{U}$$

Script *ibis_background_cor* combines the following executables:

- *ii_shadow_ubc*
- *ip_shadow_ubc*

12.6.1 *ii_shadow_ubc*

ii_shadow_ubc reads all raw detector, and corresponding efficiency, shadowgrams, and fills the detector dead zones. *ii_shadow_ubc* is run in three cases; to produce images, spectra and lightcurves. In the imaging case, for each pair of detector-efficiency shadowgrams it performs the background correction for the uniform (default) and non-uniform spatial distribution. The background normalization is calculated on the base of shadowgrams from which the pixels affected by all sources listed in the `brPif` catalog more than defined by the `brPifThreshold` parameter were removed.

As output it produces 3 shadowgrams of larger dimensions: corrected expanded shadowgram and corresponding variance and efficiency shadowgrams. Method to be applied for the pixel value interpolation in dead zones is defined by the parameter **method_int**.

Table 16: *ii_shadow_ubc* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
ii_shadow_ubc			
SCW1_BKG_I_method_cor	method_cor	integer	Method to be applied for background removal possible values: 0-2 `0` – no background correction `1` – background from IC tree is applied to the whole detector `2` – background is treated for each module separately default: `2`
SCW1_BKG_I_method_int	method_int	integer	Method to be applied for the pixel value interpolation possible values: `0` – dead zone pixels will be filled with 0 `1` – dead zone pixels will be filled with mean detector value `-1` – no dead zones default:`1`
IBIS_NoisyDetMethod	NoisyPixControl	integer	0 => no Noisy Pixel detection; 1 => use SELECT_FLAG method. default:`0`
brSrcDOL	brPif	string	DOL of the bright sources catalogue, which will be removed from background mean calculation default “ ”
SCW1_BKG_I_brPifThreshold	brPifThreshold	real	pixels with PIF value higher will be removed from background calculation. When = 1, bright PIF removal will not be performed possible values: `‘‘0 -- 1 ’’` default: `‘‘0.0001’’`
ModPixShad	ModPixShad	integer	Minimum number of non-illuminated pixels per module possible values: `100 - 500` default:`400`

12.6.2 *ip_shadow_ubc*

ip_shadow_ubc reads raw detector shadowgrams and performs the background correction. Also detector dead zones are filled at this step. Method to be applied for the pixel value interpolation in dead zones is defined by the parameter **method_int**. Expanded intensity and variance shadowgrams are produced as output.

Table 17: *ip_shadow_ubc* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
SCW1_BKG_P_picsSUnifDOL	picsSUnifDOL	string	DOL of the PICsIT Detector Uniformity model (single).
SCW1_BKG_P_picsMUnifDOL	picsMUnifDOL	string	DOL of the PICsIT Detector Uniformity model (multiple).

SCW1_BKG_P_picsSBkgDOL	picsSBkgDOL	string	DOL of the PICsIT Detector Background model (single).
SCW1_BKG_P_picsMBkgDOL	picsMBkgDOL	string	DOL of the PICsIT Detector Background model (multiple).
SCW1_BKG_P_method	method	integer	Method to use in scaling the background maps possible values: `0` – model (in cts) is normalized by the time of observation `1` – model (in cts) is normalized by the mean count value default: `0`

12.7 Catalogs

The catalog extraction selects the sources in the field of view (FOV) from the references catalog.

12.7.1 *cat_extract*

The executable *cat_extract* performs the source selection from a reference catalog. The reference catalog should have the same structure as GNRL-REFR-CAT (see Table 50 in Appendix C.5). Its DOL is passed in the parameter 'refCat'. There is no standard output catalogue for IBIS and the parameter 'outExt' must be filled.

Table 18: *cat_extract* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
CAT_refCat	ref_cat	string	DOL of the reference catalogue. default value: `''$ISDC_REF_CAT''`
OBS1_CAT_radiusMin	radiusMin	string	Low limit for the position selection. default: `''0''`
OBS1_CAT_radiusMax	radiusMax	string	High limit for the position selection. default: `''20''`
OBS1_CAT_fluxDef	fluxDef	string	Column used for flux selection. default: `''''`
OBS1_CAT_fluxMin	fluxMin	string	Low limit for flux selection (in ph/cm^2/s). default: `''''`
OBS1_CAT_fluxMax	fluxMax	string	High limit for flux selection (in ph/cm^2/s). default: `''''`
OBS1_CAT_class	class	string	Select the sources by class. default: `''''`
OBS1_CAT_date	date	string	IJD for the public data. possible values: <`0` – all public; `0` – all private; >`0` – according to 'DAY_ID' date default: -1

12.8 Image analysis

The IBIS telescope is a device based on a coded aperture imaging system. The mask chosen for IBIS is based on a cyclic replication of MURA (Modified Uniformly Redundant Array) of order 53, expanded to 95

pixels. The properties of the MURA patterns are described in the papers [11], [12].

For IBIS, the mask is about 1.8 times larger than the detector. The most important advantage of such configuration is that for a large fraction of the sky a source projects a complete pattern on the detector (the Fully Coded Field Of View, FCFOV). The part of the sky from which only a fraction of the source radiation directed towards the detector plane is modulated is called Partially Coded Field of View (PCFOV). The complete field of view of the telescope (FOV) is therefore composed by the central FCFOV of constant sensitivity and optimum image properties surrounded by the PCFOV of decreasing sensitivity. A source outside the FOV simply contributes to the background level.

Representing the mask with an array M of 1 (open elements) and 0 (opaque ones), the detector array D will be given by the convolution of the sky image S by M plus an unmodulated background array term B,

$$D = S \star M + B$$

MURA patterns have the remarkable property that their cyclic autocorrelation gives a delta function. The decoding array, G, is inverse to M where $G = 2M - 1$ (i.e., G=+1 for M=1 and G=−1 for M=0) and thus is *correlation inverse.*

With the help of array G we can reconstruct the sky:

$$S' = D \star G = S + B \star G$$

where S' differs from S only by the $B \star G$ term, which for a flat array B is a constant level that can be measured and removed.

To have a sidelobe-free response a source must be able to cast a whole basic pattern on the detector (fully coded source). To make use of all the detector area and to allow more than one source to be fully coded, the mask basic pattern is normally taken as the same size and shape of the detector and the total mask made by a cyclic repetition ($< 2 \times 2$ for rectangular mask) of the basic pattern. For such *optimum systems* a FCFOV source will always project a cyclically shifted version of the basic pattern, and correlating the detector image with the G decoding array will provide a sidelobe-free peak with position-invariant shape at the source position.

A source in the PCFOV will instead cast an incomplete pattern and its contribution cannot be a-priori subtracted and will produce secondary lobes (coding noise). On the other hand the modulated radiation from PC sources can be reconstructed by extending the correlation procedure, with a proper normalization, to the PCFOV.

URA masks also minimize the statistical errors of the reconstructed peaks. Since $V = G^2 \star D = \Sigma D$ the variance associated with each reconstructed sky image pixel is constant in the FCFOV and equal to the total counts recorded by the detector; therefore the source signal to noise is simply

$$\frac{S}{N} = \frac{C_S}{\sqrt{C_S + C_B}}$$

where C_S and C_B are source and background counts. These masks also have nearly equal number of transparent and opaque elements and therefore offer minimum statistical error in condition of high background (typical for the γ-ray domain). However the sensitivity also depends on the detector spatial resolution and an *imaging efficiency* factor must be applied to this maximum S/N to account for this effect.

Discrete cross-correlation to compute sky and variance images can be written

$$S_{ij} = \sum_{kl} G_{i+k,j+l} D_{kl} \quad ; \quad V_{ij} = \sum_{kl} G^2_{i+k,j+l} D_{kl}$$

where Poisson statistics was assumed. This standard deconvolution in FCFOV can be extended in the PCFOV by extending the correlation of the decoding array G with the detector array D in a non-cyclic form, padding G with 0 elements. Since only the detector section modulated by the PC source is used to

reconstruct the signal, the statistical error at the source position and significance of the ghost peaks are minimized. However to ensure a flat image in the absence of sources, detector pixels which for a given sky position correspond to opaque mask elements must be balanced before subtraction with the factor $b = n^+/n^-$, where n_+ is the number of pixels corresponding to transparent elements and n_- to opaque ones for that given sky position. This can be written

$$S_{ij} = \sum_k G^+_{i+k,j+l} W_{kl} D_{kl} - B_{ij} \sum_{kl} G^-_{i+k,j+l} W_{kl} D_{kl}$$

where the decoding arrays are obtained from the mask M by $G^+ = M$ and $G^- = 1 - M$, then padded with 0's outside mask region, and where the sum is performed over all detector elements. In the FCFOV we obtain the same result as the standard cross-correlation. To consider effects such as satellite drift corrections (see [7]), dead areas or other specific conditions, a weighting array W is used to weigh properly the detector array before correlating it with the G arrays. The balance array is

$$B_{ij} = \frac{\sum_{kl} G^+_{i+k,j+l} W_{kl}}{\sum_{kl} G^-_{i+k,j+l} W_{kl}}$$

The variance, which is not constant outside the FCFOV, is computed accordingly

$$V_{ij} = \sum D_{kl} \left(G^+_{i+k,j+l} W_{kl}\right)^2$$
$$+ B^2_{ij} \sum D_{kl} \left(G^-_{i+k,j+l} W_{kl}\right)^2$$

since the cross-terms G^+G^- vanish. Note however that when the weights W_{kl} refer to the same pixel in D, the terms $G\,W$ must be summed before squaring (see [7]). The varying effective area can be calculated by a similar formula and used to renormalize, after background subtraction to FCFOV count rates. All this can be performed for sampling finer than 1 pixel per mask element and using a G array convolved with detector the PSF in order to optimize S/N for point sources, with corresponding normalizations. This procedure can be carried out with a fast algorithm by reducing previous formulae to a set of correlations computed by FFT.
The on-axis System Point Spread Function (SPSF) on the whole FOV for an optimum system and PSF deconvolution is shown in Fig. 32. Note the peak and flat level in the central FCFOV, the secondary lobes (coding noise) in the PCFOV and the 8 main ghosts of the source peak in the PCFOV located at distances, from the source, which are a multiple of the basic pattern.

The average Point Source Location Error (PSLE) for an optimum coded aperture system with a defined SPSF depends on the source signal to noise ratio (S/N) as following

$$PSLE \div \frac{1}{R\ (S/N)}$$

The IBIS/ISGRI telescope, assuming no error in pointing axis reconstruction or other systematic effects, can locate a 30 σ point-like source at better than $\pm\ 1'$. Absolute error in attitude reconstruction for INTEGRAL is expected to be $< 20''$.

In a standard analysis, IBIS events or histograms are binned in detector images, which are then corrected for detector and background non-uniformity [7] and then processed by an iterative algorithm which decodes, cleans and composes sky images. For each detector image a sky image and its variance are obtained using the deconvolution procedure, and then iteratively searched for sources and cleaned of the source side lobes. In this iterative process the source peaks are fitted with the bi-dimensional Gaussian and finely located. Then the source contribution to the image is modeled in detail and subtracted. The images are rotated, projected and summed after being weighted with the variance, and then searched for further contributions.

More details can be found in [8], [9]

12.8.1 *ii_skyimage*

ii_skyimage deconvolves shadowgrams in the given energy bands for each Science Window, using the balanced cross-correlation method described above. The energy bands should be either the same as in the *ibis_binning*

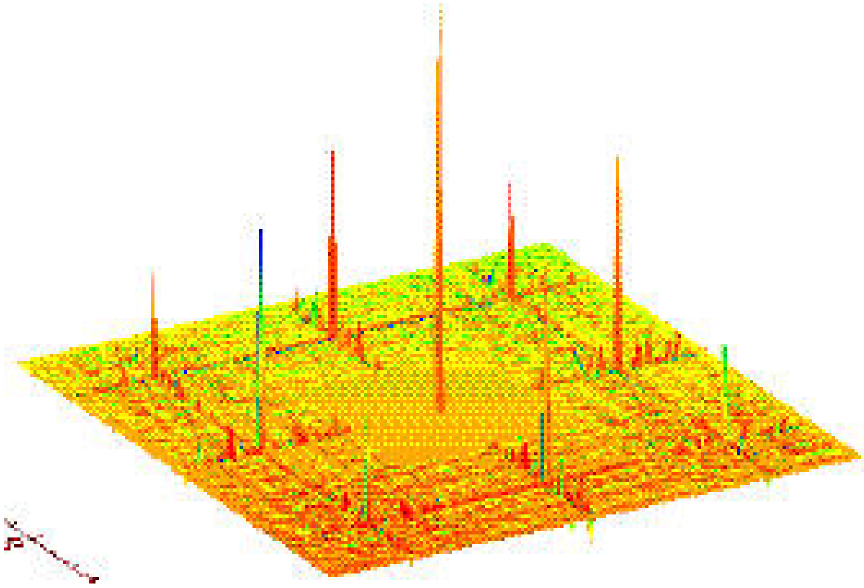


Figure 32: SPSF for the IBIS/ISGRI telescope.

or the bands created by the merger of those bands. The deconvolution is weighted by efficiency. Thus the weighting array W contains efficiency for normal pixels and 0 elsewhere.

A search for sources is performed in the deconvolved image. The list of sources found is created and the images are cleaned from the PSF ghosts.

In the deconvolved image, the pixel value at the source position is the total source flux in cts/sec units. It is calculated as if the source was in the fully coded field of view (FCFOV). One should NOT do any integration in the source region to estimate the flux from the image. The source flux estimation given in the source list is slightly different as the source flux is given at the fitted source position.

For each output energy band all images of the same type are combined into one mosaic image.

A search for the sources in the mosaic image is then performed and a list of sources found is created.

The current version of software allows you to create the mosaic separately from the main analysis (see Section 7.1 for the details).

The Parameter **PixSpread** sets whether the source flux is spread, or not in the mosaic image.

No-spread mosaic puts the whole input pixel count into one output map pixel. It permits better flux and signal-to-noise ratio approximation as each pixel count and variance is summed without any error. On the other hand it can give some undesired effects as double source peaks because of binning.

In the spread mosaic the input pixel count is spread between some number of output map pixels. This method is better for source position recovery but not so good for source flux estimation, because of the source peak height reduction. Furthermore, neighbouring pixels in the individual Science Window images are correlated and correct spread variance calculations should take this into account. But as exact calculation of the covariance matrix is too heavy for on-line analysis, the approximative variance formula is used.

The *ii_skyimage* parameters are given in the Table 19.

Table 19: *ii_skyimage* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description

OBS1_mask	mask	string	DOL of the MASK pattern fits file default: ‘‘ ’’
OBS1_deco	deco	string	DOL of the projected decoding pattern fits file default: ‘‘ ’’
OBS1_DataMode	DataMode	integer	Data Simulation mode possible values: `0` – shadowgrams are treated `1` – shadowgrams are simulated default: `0`
OBS1_SearchMode	SearchMode	integer	Source search mode possible values: `0` – search for all significant excesses `1` – search for all catalog sources `2` – search for K sources,where K= **ToSearch** `3` – search for all catalog sources plus K significant excesses,where K = **ToSearch** Parameter **SearchMode**= $1, 3$ concerns only images at a Science Window level. In the mosaic image *ii_skyimage* always looks for K = **ToSearch** sources. default: `3`
OBS1_ToSearch	ToSearch	integer	Number of sources to be looked for. default: `50`
OBS1_CleanMode	CleanMode	integer	Ghost cleaning possible values: `1` – source model lobes are subtracted from the deconvolved image `-1` – no subtraction default: `1`
tungAtt	tungAtt	string	DOL of the Tungsten attenuation length fits file default: ‘‘ ’’
aluAtt	aluAtt	string	DOL of the Aluminium attenuation length fits file default: ‘‘ ’’
leadAtt	leadAtt	string	DOL of the Lead attenuation length fits file default: ‘‘ ’’
OBS1_ScwType	ScwType	string	Type of Science Window to be treated possible values: ‘‘`POINTING`’’, ‘‘`SLEW`’’, ‘‘`OTHER`’’, ‘‘`ANY`’’ default: ‘‘`POINTING`’’
OBS1_DoPart2	DoPart2	integer	if `1` or absent → do mosaic part. Set it to `0` if you don’t want to produce a mosaic image. default: `1`
OBS1_MapAlpha	MapAlpha	real	Mosaic map center [deg] default: `""`
OBS1_MapDelta	MapDelta	real	Mosaic map center [deg] default: `""`
OBS1_MapSize	MapSize	real	Mosaic map radius [deg] default: `40.0`
OBS1_PixSpread	PixSpread	integer	`0` – no flux spread in mosaic default: `1`
OBS1_MinCatSouSnr	MinCatSouSnr	real	Software detects the catalog source only if its signal-to-noise ratio is higher then this value. This parameter has no meaning for the Science Window level images if **OBS1_SearchMode**=*1,3*. default: `6`

OBS1_MinNewSouSnr	MinNewSouSnr	real	Software detects a new source only if its signal-to-noise ratio is higher then this value. default: `7`
OBS1_SouFit	SouFit	integer	0 for fitting source position and 1 for fixed source position in ScW fit default: `0`
OBS1_ExtenType	ExtenType	integer	exposure/residual maps possible values: 0 or absent - Residual Map at 4*n-th extension of ima idx 1 - Exposure Map only at 4*n-th extension of ima idx 2 - both Residual and Exposure maps at 4,5*n-th extension of ima idx 3 - one Expo map at the end of ima idx if ExtenType=1,2,3 the Mosaic Exposure Map is at the 4*n-th extension of mosa idx (instead of the on-time map) default: 2
OBS1_NegModels	NegModels	integer	0 or absent for no negative models ; 1 - negative models default: `0`
OBS1_FastOpen	FastOpen	integer	if 1 then no CommonPreparePars default: `1`

12.8.2 *sumhist*

PICsIT needs long integration times to produce a good image. Thus if *INTEGRAL* was stable during several Science Windows it is recommended to sum up PICsIT shadowgrams before the deconvolution. If the *ibis_science_analysis* parameter **staring** is set to `yes`, then *sumhist* checks that *INTEGRAL* was stable during all the Science Windows within the given limits and sums all the PICsIT shadowgrams into one.

Table 20: *sumhist* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
tolerance	tol	real	tolerance for staring default: `0.0001`

12.8.3 *ghost_busters*

We have found that some regions of the mask have glue deposits in the void. The shadowgrams of strong sources near those glue deposits are not fully correct and create artifacts in the images (mostly ghosts). In the new ghost_buster processing, affected pixels are killed from the shadowgrams for strong sources (sourcecat="$ISDC_REF_CAT[ISGRI_FLAG2==5])". If you include too many sources (by changing the default catalog parameter SCW1_BKG_buster_src in ibis_science_analysis) in this ghost buster algorithm you will also kill too many signal pixels, so only include very bright sources and do this if you see artifacts in very deep mosaics.

Table 21: *ghost_busters* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
SCW1_BKG_busters_mask	maskmode	string	Model of the mask areas to be ignored.
SCW1_BGK_busters_src	sourcecat	string	Input source catalog.

12.8.4 *ip_skyimage*

ip_skyimage performs deconvolution of the shadowgrams with the use of the balanced cross-correlation method described above, for more detailed explanations see Goldwurm et al. 2003, [10]. The weighting array W corresponds to the efficiency map. The values of the decoding G-array are taken from the IC file.

The *ip_skyimage* parameters included into the main script are given in the Table 22.

Table 22: *ip_skyimage* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
PICSIT_detThr	detThr	real	Detection threshold for the automatic source location.
PICSIT_inCorVar	inCorVar	integer	Corrected variance to be used [0=NO; 1=YES]
PICSIT_outVarian	outVarian	integer	Variance maps in output [0=NO; 1=YES]
PICSIT_deco	maskDeco	string	DOL of the decoding pattern of PICsIT mask.

12.9 Spectral Analysis

12.9.1 *ii_spectra_extract*

ii_spectra_extract reads the shadowgram for each input energy band.

For each output energy band, it :

- creates a model (PIF) for each source. The Pixel Illumination Factor (PIF) is a number between 0 and 1, which expresses the theoretical degree of illumination of each pixel in the detector plane of a coded-mask instrument. The PIF is calculated on the basis of the position and is normalized to 1 cts/pix source.
- fits all source intensities together with a background estimate to the data (shadowgrams)
- writes all source estimates to the spectra index

Finally, *ii_spectra_extract* writes to the PIF index one mean model per input catalogue source.

There is a possibility to chose one of several fitting metods. The recommended method is iterative imaging-like Least Squares fit (method 6). In the first step of this method the fit of each source separately in its coded zone is done. Next, in the second step, the calculation of reciprocal source contributions is performed. The first step is aimed to decrease the background modelling error influence. The second one, corresponds to the ghost cleaning. Please note that Maximum Likelihood method is not working in standard OSA configuration (with isdcmath package).

In the fitting process the source positions are considered to be exact. If the input background map is given , then the background map fitting is performed.

Table 23: *ii_spectra_extract* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
SCW2_ISPE_idx_isgrResp	idx_isgrResp	string	DOL of the index of ISGRI response matrices default: ‘‘’’
SCW2_ISPE_DataMode	DataMode	integer	Data Simulation mode possible values: 0 real data 1 simulated data default: 0

SCW2_ISPE_MethodInt	MethodInt	integer	Method to be applied for the pixel value interpolation default: 1
SCW2_ISPE_MethodFit	MethodFit	integer	Method to be applied for background and source intensity fitting possible values: MethodFit = 0 ==> CHI2 MethodFit = 1 ==> Maximum Likelihood MethodFit = 2 ==> Least Squares MethodFit = 6 ==> iterative imaging-like Least Squares fit default: 6
SCW2_ISPE_isgrUnifDol	isgrUnifDol	string	DOL of the ISGRI detector uniformity default: ‘‘’’
SCW2_ISPE_isgrBkgDol	isgrBkgDol	string	DOL of the ISGRI background model default: ‘‘’’
tungAtt	tungAtt	string	DOL of the Tungsten attenuation length fits file default: ‘‘’’
aluAtt	aluAtt	string	DOL of the Aluminium attenuation length fits file default: ‘‘’’
leadAtt	leadAtt	string	DOL of the Lead attenuation length fits file default: ‘‘ ’’
rebinned_corrDol_spe	corrDol	string	DOL of the rebinned ISGRI off-axis corrections for spectrum default: ‘‘ ’’

12.9.2 *ip_spectra_extract*

ip_spectra_extract reads the shadowgram for each input energy band and the name and coordinates of the source for which a spectrum will be extracted. Since PICsIT operates in an energy range with a few sources, this executable works for one source only.

For each output energy band, it:

- creates a model (PIF) for the source. The Pixel Illumination Factor (PIF) is a number between 0 and 1, which expresses the theoretical degree of illumination of each pixel in the detector plane of a coded-mask instrument. The PIF is calculated on the basis of the position and is normalized to 1 cts/pix source. The present PIF is purely geometrical.
- by taking into account that there is one single source with ideal PIF, than the Equation at pag. L228 of Goldwurm et al. (2003) can be simplified. In this case, all the pixel counts, depending on the PIF values, are due to the selected single source. That is, deviations from the average counts in the observed shadowgrams are due entirely to the source.
- writes the source count rates in the selected energy bands and the PIF.

Table 24: *ip_spectra_extract* parameters included into the main script.

Name (in the main script)	Name (executable)	Type	Description
PICSIT_source_name	inName	string	Name of the source for extracting PICsIT spectrum default: ‘‘ ’’
PICSIT_source_RA	inRA	real	RA of the source for extracting PICsIT spectrum default: ‘‘ ’’
PICSIT_source_DEC	inDEC	real	DEC of the source for extracting PICsIT spectrum default: ‘‘ ’’

PICSIT_source_inOG	inOG	string	DOL of the input Observation group default: ‘‘ ’’
PICSIT_source_Shadow	inShadow	string	DOL of the input Shadowgrams default: ‘‘ ’’
PICSIT_source_mask	maskG	string	DOL of the IBIS mask model default: ‘‘ ’’
PICSIT_source_outOG	outOG	string	DOL of the output Observation group default: ‘‘ ’’
PICSIT_source_outPIF	outpif	string	DOL of the PIF index default: ‘‘ ’’
PICSIT_source_outSpec	outSpec	string	DOL of the output spectra default: ‘‘ ’’

12.10 Timing Analysis

12.10.1 *ii_lc_extract*

ii_lc_extract performs similar tasks to *ii_spectra_extract*, dealing with time bins instead of spectra ones.

Table 25: Parameters for the *ii_lc_extract*.

Name (in the main script)	Name (executable)	Type	Description
ISGRI_mask	mask	string	DOL of the MASK pattern fits file default: ‘‘ ’’
SCW2_cat_for_extract	inCat	string	DOL of the subset of the source catalogue default: ‘‘ ’’
SCW2_ISPE_DataMode	DataMode	integer	Data Simulation mode DataMode = 0 ==> OG shadowgrams are treated 1 ==> shadowgrams are simulated
SCW2_ISPE_MethodInt	MethodInt	integer	Method to be applied for the pixel value interpolation This parameter should have the same value as in the *ii_shadow_build* In the source model procedure MethodInt= 0 ==> dead zone pixels will be filled with 0 1 ==> dead zone pixels will be filled with mean detector value -1 ==> no dead zones
SCW2_ISPE_MethodFit	MethodFit	integer	Method to be applied for background and source intensity fitting default: 6
SCW2_ISPE_isgrUnifDol	isgrUnifDol	string	DOL of the ISGRI detector uniformity default: ‘‘ ’’
SCW2_BKG_I_isgrBkgDol	isgrBkgDol	string	DOL of the ISGRI background model default: ‘‘ ’’
tungAtt	tungAtt	string	DOL of the Tungsten attenuation length fits file default: ‘‘’’
aluAtt	aluAtt	string	DOL of the Aluminium attenuation length fits file default: ‘‘’’
leadAtt	leadAtt	string	DOL of the Lead attenuation length fits file default: ‘‘ ’’
rebinned_corrDol_lc	corrDol	string	DOL of the rebinned ISGRI off-axis corrections for lc default: ‘‘ ’’

SCW2_BKG_I_brPifThreshold	brPifThreshold	real	pixels with PIF value higher will be removed from background calculation. When = 1, bright PIF removal will not be performed possible values: `‘‘0 -- 1 ’’` default: `‘‘0.1’’`
ModPixShad	ModPixShad	integer	Minimum number of non-illuminated pixels per module possible values: `100 - 500` default:`400`

12.10.2 *ip_st_lc_extract*

This program builds PICsIT detector light curves and related errors starting from spectral timing data of PICsIT. No parameters of this executable are included into the parameter file of the main script *ibis_science_analysis*.

12.11 Summing up the results

12.11.1 *ip_skymosaic*

ip_skymosaic creates a mosaic image of all the science windows within an Observation Group, and performs a source location

Table 26: *ip_skymosaic* parameters included into the main script.

Name In the main script	Name	Type	Description
OBS2_detThr	detThr	real	Detection Threshold in Sigmas default: `3`
OBS2_projSel	projSel	string	Selection of projection default: STG

12.12 Tools not included in the pipeline

12.12.1 *mosaic_spec*

mosaic_spec is a tool that extracts a count spectrum at a given sky position from a set of mosaic sky images. Intensity could be measured in a specific or in the most significant pixel or derived from a gaussian fit with free or frozen position and width.

Note that if the significance of the point is less than five, then *mosaic_spec* will assign a non-zero value to its quality flag.

Table 27: *mosaic_spec* parameters.

Name	Type	Description
DOL_inp	string	DOL of the group containing the input images. default: “”
DOL_out	string	DOL of the group containing the output images. default: “”
EXTNAME	string	EXTNAME of the input images. default: “`ISGR-MOSA-IMA`”

DOL_idx	string	DOL of the index of input images. default: “”
DOL_spec	string	DOL of the index of output images. default: “”
ximg	real	”Enter source x coordinate in pixels. possible values: `0 -- 10000` default: `0`
yimg	real	Enter source y coordinate in pixels. possible values: `0 -- 10000` default: `0`
ra	real	Enter source RA possible values: `-180 -- 360` default: `0`
dec	real	Enter source DEC possible values: `-90 -- 90` default: `0`
posmode	integer	Position fitting mode posible values: -1: freeze position as input; 0: position is always left free; 1: freeze position as found in first energy band default: `0`
widthmode	integer	Gaussian width fitting mode posible values: -1: radial width frozen as input 0: radial width left free 1: x and y width left free 2: x and y width fixed as found for first band default: `0`
psf	real	HWHM of PSF in arcmin possible values: `0.1 -- 20` default: `6`
size	integer	”Enter source box half size in pixels on which the fit is performed. possible values: `1 -- 1000` default: `20`
back	boolean	Should a constant intensity background be considered in the fit default: `no`
allEnergies	boolean	Enter if the program has to select automatically energies default: `yes`
emin	string	Enter vector of energy bands minimum. default: “`25 30 40`”
emax	string	Enter vector of energy bands maximum. default: “`30 40 60`”

chatty	integer	Enter reporting-level (10: includes MINUIT log). possible values: `0 -- 10` default: `4`

12.12.2 *ii_light*

For every PIF found in the Science Window, *ii_light* extracts simultaneously a lightcurve for each source and one light curve for background in all specified energy bands. Dead pixel, data gaps, off-axis correction, energy low threshold and illumination factors are taken into account.

The method used is a fit of hyperplane through the cloud of points formed by the number of counts versus PIF fraction for every source. In the case of one source this is just a linear regression. The intercept gives the background, while the slope gives the flux of the source in one fully illuminated pixel. Normalization is a number of counts in a perfect detector if source is on axis (half of all detector pixels illuminated).

There are two methods to extract the flux for one source:
1. the source flux is extracted considering the source alone.
2. the source flux is extracted taking into account that other sources are in the FOV.
For a faint source in the PCFOV it is important to take into account all bright (typically larger than 0.1 Crab) sources in the entire FOV, whereas to analyse a faint source in the FCFOV it is necessary to consider only bright sources in the PCFOV.

The sources that need to be considered simultaneously for the lightcurve extraction are the "essential sources" (the DOL to the list is stored in *source_selectDol* parameter). The list however should include the lower possible number of sources (defined through the *maxessential* parameter). This is due to the fact that extracting simultaneously the flux of many (N) sources takes a longer (of order N^3) computation time than extracting a source alone.

Care has been taken so that the output structure is compatible with HEASARC tools.
There are no limits on the size of time bin (up to 0.1s), and number of energy bands.

ii_light is not the official lightcurve extraction tool and should be used mainly to check relative variability of bright sources within a given Science Window, rather than for a long term absolute flux estimate. An extended comparison between the official lightcurve extraction tool (ii_lc_extract) and ii_light can be found in Grinberg et al. 2011 [15]

Table 28: *ii_light* parameters.

Name	Type	Description
inSwg	string	DOL of the input Observation Group. default: " "
outSwg	string	DOL of the output Observation Group default: " "
outLC	string	Dol of the light curves file default: " "
context	string	DOL of the context where we can find low threshold default: " "
GTIname	string	Name of the GTI to be applied default: " "
select	string	Event selection condition default: " "

pifDOL	string	Dol of the PIF (or of the index of PIF) default: “ ”
deadDOL	string	Dol of the ISGR-DEAD-SCP” default: “ ”
corrDol	string	DOL of the isgri off-axis corrections default: “ ”
backDol	string	DOL of the background maps default: “ ”
delta_t	real	Time bin in seconds. possible values: `0.1` – `10000` default: `100`
num_e	integer	Number of energy channels possible values: `1` – `10` default: `2`
e_min	string	List of low energy boundaries default: “`15 40`”
e_max	string	List of high energy boundaries default: “`40 300`”
onlydet	boolean	Do only detector light curve? default: NO
idxSwitch	string	DOL of the index of pixels switches list. default: “ ”
idxNoise	string	DOL of the index of noisy pixels default: “ ”
evttype	integer	Type of data 0=ISGRI; comprised between 0 and 5 default: `0`
source_selectDol	string	DOL containing column NAME of essential sources default: “ ”
maxessential	integer	Max number of essential sources (between 1 and 300) default: `3`
cleanobt	boolean	Discard equal time photons default: NO
chatter	integer	Verbosity of output. (0: min verbosity; 5: max verbosity) default: `2`

A Low Level Processing Data Products

A.1 Raw Data

Table 29 summarizes all Data Structures with the raw events measured by IBIS. The description of these Data Structures can be found below.

Table 29: List of IBIS ****-****-RAW Data Structures

Observing mode	Type of event	Raw Data Structure	Prepared Data Structure	Corrected Data Structure
Photon-by-photon	ISGRI [I]	ISGR-EVTS-RAW	ISGR-EVTS-PRP	ISGR-EVTS-COR
	PICsIT single [PS]	PICS-SGLE-RAW	PICS-SGLE-PRP	PICS-SGLE-COR
	PICsIT multiple [PM]	PICS-MULE-RAW	PICS-MULE-PRP	PICS-MULE-COR
	Compton single [CS]	COMP-SGLE-RAW	COMP-SGLE-PRP	COMP-SGLE-COR
	Compton multiple [CM]	COMP-MULE-RAW	COMP-MULE-PRP	COMP-MULE-COR
Spectral Imaging Histogram	PICsIT single	PICS-SISH-RAW		
	PICsIT multiple	PICS-SIMH-RAW		
Spectral Timing Histogram		PICS-SPTI-RAW	PICS-SPTI-PRP	

A.1.1 Photon-by-photon mode

In this mode Data Structures with similar contents are created for all types of events. Data Structures with names finished by **RAW** contain the information about the event itself. See Table 30 for details.

Data Structures with names finished by **PRW** and **SRW** contain some technical information about the structure of the telemetry packet and the Local On-Board Time (LOBT) of the events.

Table 30: Contest of Photon-by-Photon Mode Raw Data.

Column Name	Description	Event Type
DELTA_TIME	Delta time to previous event	I, PS, PM
RISE_TIME	Event rise-time (describes the shape of the registered pulse)	I, CS, CM
ISGRI_PHA	Pulse height in the ISGRI layer	I, CS, CM
PICSIT_PHA	Pulse height in the PICsIT layer	PS, PM, CS, CM
ISGRI_Y	Y location in the ISGRI layer	I, CS, CM
ISGRI_Z	Z location in the ISGRI layer	I, CS, CM
PICSIT_Y	Y location in the PICsIT layer	PS, PM, CS, CM
PICSIT_Z	Z location in the PICsIT layer	PS, PM, CS, CM
CAL_FLAG	Calibration flag (1 - when neither of the events are calibration ones, only in this case the event is used in the following analysis.)	CS, CM
TIME_TAG	Compton delta time to previous event	CS, CM
DUMMY_COUNTER	Dummy counter - some technical information used to reconstruct the On-Board Time.	CS, CM

A.1.2 PICsIT Standard Mode

In this mode the information is accumulated on-board and transmitted to the Earth in a kind of histograms. **Spectral-image** histograms are written to the Data Structures **PICS-SISH-RAW** and **PICS-SIMH-RAW** for single and multiple events correspondingly. These Data Structures represent a three dimensional

table with total counts during integration time. The axes of the grid are directed along the following axes: AXIS 1 – Channel number, AXIS 2 – Y position, AXIS 3 – Z position.

Spectral-timing histograms are written to the Data Structure **PICS-SPTI-RAW** and contain the information on the number of the events accumulated in the up to 8 energy bins during a given time amount, see Table 31

Table 31: Content of PICS-SPTI-RAW Data Structure.

Column Name	Description
CELL_1	1st cell of the spectrum
CELL_2	2nd cell of the spectrum
CELL_3	3rd cell of the spectrum
CELL_4	4th cell of the spectrum
CELL_5	5th cell of the spectrum
CELL_6	6th cell of the spectrum
CELL_7	7th cell of the spectrum
CELL_8	8th cell of the spectrum

The local on-board time and the channels definitions can be found in the **PICS-SPTI-PRW** Data Structure.

A.2 Prepared Data

The main task of the Science Window Pipeline is to prepare raw data for the following Scientific Analysis. It converts the housekeeping parameters into the physical units and makes some corrections and transformations of the raw data that are not included in Pre-Processing. The Summary of all the prepared Data Structures with scientific information can be found in Table 29. All these Data Structures has the only column **OB_TIME** with the on-board time.

B Instrument Characteristics used in Data Analysis.

B.1 Noisy Pixels

It is possible that with time some of the pixels of the detector start to produce an output not triggered by an income photon, i.e. to become "noisy". If the particular pixel countrate is too high relatively to the module countrate, then the on-board electronics switch it off. In ISGRI the noisy pixels can recover after being switched off for some time and disabled pixels are periodically reset to check their status. Data Structure **ISGR-SWIT-STA** with the list of noisy pixel switches in the ISGRI detector layer is produced during the pre-processing basing on the rise-time information and pixels light-curves transmitted, see Table 32.

Table 32: Content of ISGR-SWIT-STA Data Structure

Column Name	Description
ISGRI_Y	Y location in the ISGRI layer
ISGRI_Z	Z location in the ISGRI layer
OBT_DETECT	First time when the pixel can be declared as noisy
OBT_SWITCH	OBT when the pixel is switched off
NOIS_FLAG	Noisy flag

In PICsIT pixels cannot be recovered that easy. A PICsIT pixel will remain off once killed. Only if half of the detector (or so) will be off pixels will be attempted to turn on.The history of the disable pixels can be found in Data Structure **PICS-FALT-STA**, see Table 33.

Table 33: Content of PICS-FALT-STA Data Structure

Column Name	Description
PICSIT_Y	Y location in the PICsIT layer
PICSIT_Z	Z location in the PICsIT layer
OBT_DETECT	First Time when the pixel can be declared as noisy
OBT_FAULT	OBT when the pixel is switched off
FALT_FLAG	FLAG reflecting the changes of the pixel status. Possible values are ON-OFF and OFF-ON

B.2 Calibration Corrections

B.2.1 ISGRI

In OSA10 the following corrections to the pulse height gains and offsets have been applied (see Section 12.1):

- For E<50 keV the gain is modeled with a linear function

$$\begin{cases} gPH_1 &= gPH_{1,0} + a_g \times \mathrm{revol} = 2.04(7) - 6.(1) \cdot 10^{-4} \times \mathrm{revol} \\ oPH_1 &= oPH_{1,0} = -5.65(5) \end{cases}$$

  The parameters needed to describe the gain for E< 50 keV are hard-coded in OSA (ibis_isgr_energy and ii_shadow_build).

- For E>50 keV the pulse height gain and offset are modeled as a function of the rise time (rt) and revolution number:

$$\begin{cases} gPH_2(\mathrm{rt}) &= gPH_{2,0}(\mathrm{rt}) + b(\mathrm{rt}) \times \mathrm{revol} \\ oPH_2(\mathrm{rt}) &= oPH_{2,0}(\mathrm{rt}) + c(\mathrm{rt}) \times \mathrm{revol} + d(\mathrm{rt}) \times \mathrm{revol}^2 \end{cases}$$

  The values for the gain and offset as a function of the rise time are delivered as calibration files in two new IC tables, ISGR_GAIN_MOD and ISGR_OFF2_MOD.

Table 34: Content of ISGR-GAIN-MOD Data Structure

Column Name	Description
PAR1	Parameter 1 for energy gain correction
PAR2	Parameter 2 for energy gain correction

Table 35: Content of ISGR-OFF2-MOD Data Structure

Column Name	Description
PAR1	Parameter 1 for energy offset correction
PAR2	Parameter 2 for energy offset correction
PAR3	Parameter 3 for energy offset correction

To perform a rise-time correction for each raw ISGRI event, the Rise-Time correction table is used (**ISGR-RISE-MOD** Data Structure, see details in Table 36). This table is composed of NUM_ENER lines, giving for each incident energy the correction factor **corr** for a given value of the rise-time.

Table 36: Content of ISGR-RISE-MOD Data Structure

Column Name	Description
ENERGY	Energy at which the gain-offset relationship is measured
CHANNEL	Channel at which the gain-offset relationship is measured
CORR	Rise-time correction for a given rise-time value (0-127)

The rise-time itself and the gain offset are kept at the **ISGR-OFFS-MOD** Data Structure.

Table 37: Content of ISGR-OFFS-MOD Data Structure

Column Name	Description
AGAIN	Amplitude Gain
AOFFSET	Amplitude Offset
RTGAIN	Risetime Gain
RTOFFSET	Risetime Offset
PIXTYPE	Pixel type

B.2.2 PICsIT

Data Structure **PICS-ENER-MOD** is created by the PICsIT Automatic Calibration Analysis and contains parameters for gain and offset correction of PICsIT raw events. Presently, all the gain values are set to 1 and all the offset values are set to 0. This means that OSA makes use only of the equalization performed onboard by using HEPI-LUT. No other corrections are applied.

Content of this Data Structure is given in the Table 38. The values given in the Table 38 are normalized to the average values, given by the keywords **AVGAIN** for average gain value (in units keV/channel) and **AVOFFSET** for average offset value (in keV).

Table 38: Content of PICS-ENER-MOD Data Structure

Column Name	Description
PICSIT_Y	Y location in the PICsIT layer
PICSIT_Z	Z location in the PICsIT layer
GAIN	Gain
OFFSET	Offset

B.3 Limit Tables

Instrument GTIs depending on HK and other parameters are defined by a limit in a limit table **IBIS-GOOD-LIM**, see details in Table 39.

Table 39: Content of **IBIS-GOOD-LIM** limit table.

Column Name	Description
PAR_NAME	Parameter name
OBT_START	Start of validity of the limit values
OBT_END	End of validity of the limit values
MIN_VAL	Minimum value allowed
MAX_VAL	Maximum value allowed
GTI_NAME	Name of the group to which the parameter belongs
SUB_ASSEMBLY	Identifier of the instrument sub-assembly
CHECK_MODE	Modes in which the parameters must be checked

B.4 Instrument Background

Table 40 lists Data Structures with the instrument background models.

Table 40: Instrument Background Model Data Structures.

Data Structure	Description
ISGR-BACK-BKG	ISGRI instrument background array.
ISGR-UNIF-BKG	ISGRI instrument background uniformity array.
PICS-SBAC-BKG	PICsIT instrument background array for single events.
PICS-SUNI-BKG	PICsIT instrument background uniformity array for single events.
PICS-MBAC-BKG	PICsIT instrument background array for multiple events.
PICS-MUNI-BKG	PICsIT instrument background uniformity array for multiple events.
COMP-SBAC-BKG	COMPTON instrument background array for single events.
COMP-SUNI-BKG	COMPTON instrument background uniformity array for single events.
COMP-MBAC-BKG	COMPTON instrument background array for multiple events.
COMP-MUNI-BKG	COMPTON instrument background uniformity array for multiple events.

For each type of Data Structures in Table 40 there is an Index. Its content is given in Table 41. Two last columns (RISE_MIN and RISE_MAX) are present only in **ISGR-BACK-BKG-IDX** and **ISGR-UNIF-BKG-IDX** Indexes.

Table 41: Content of Indexes for Table 40 Data Structures.

Column Name	Description
VERSION	Version of the instrument characteristic file
VSTART	Start of validity time in IJD
VSTOP	End of validity time in IJD
E_MIN	Lower bound of the energy range
E_MAX	Upper bound of the energy range
EXPOSURE	Effective exposure time
VETO_THR	Veto threshold
METH_BKG	Method used to produce this background image
RISE_MIN	Minimum event rise time channel 0-127
RISE_MAX	Maximum event rise time channel 0-127

C Science Data Products

C.1 ibis_correction

This script converts photons energy into keV with the help of the calibration data. Table 42 summarises the output Data Structures.

Table 42: List of Data Structures produced at COR level

Observing mode	Type of event	Corrected Data Structure
Photon –	ISGRI [I]	ISGR-EVTS-COR
	PICsIT single [PS]	PICS-SGLE-COR
by –	PICsIT multiple [PM]	PICS-MULE-COR
	Compton single [CS]	COMP-SGLE-COR
photon	Compton multiple [CM]	COMP-MULE-COR

The content of the level COR Data Structures for the photon-by-photon mode is given in Table 43. The abbreviations for the events types are taken from Table 42.

Table 43: Content the level COR Data Structures for the photon-by-photon mode.

Column Name	Description	Event Type
ISGRI_PI	corrected rise-time for ISGRI	I,CS,CM
ISGRI_ENERGY	Deposited energy in the ISGRI layer	I,CS,CM
PICSIT_ENERGY	Deposited energy in the PICsIT layer	PS,PM,CS,CM
SELECT_FLAG	Selection flag - shows whether the event was noisy (1) or not (0).	I,CS,CM

C.2 *ibis_gti*

This script builds Good Time Intervals from housekeeping data, information about satellite stability, and data gaps. The resulted GTIs are written to the **IBIS-GNRL-GTI**, see details in Table 44. Index of all **IBIS-GNRL-GTI** Data Structures for all categories is written to the **IBIS-GNRL-GTI-IDX**.

Table 44: Content of **IBIS-GNRL-GTI** Data Structures.

Column Name	Description
OBT_START	On-board time of start of the GTI
OBT_END	On-board time of end of the GTI

C.3 *ibis_dead*

This script calculates the dead times for ISGRI, PICsIT and Compton cases.

The results of the executables are written to Data Structures **ISGR-DEAD-SCP**, **PICS-DEAD-SCP** and **COMP-DEAD-SCP**. This Data Structures contains the information about the on-board time and the dead time for each module, see Tables 45 – 47.

Table 45: Content of **ISGR-DEAD-SCP** Data Structures.

Column Name	Description
OB_TIME	On-board time
II_DEADTIME_0	Dead time for module 0

II_DEADTIME_1	Dead time for module 1
II_DEADTIME_2	Dead time for module 2
II_DEADTIME_3	Dead time for module 3
II_DEADTIME_4	Dead time for module 4
II_DEADTIME_5	Dead time for module 5
II_DEADTIME_6	Dead time for module 6
II_DEADTIME_7	Dead time for module 7

Table 46: Content of **PICS-DEAD-SCP** Data Structures.

Column Name	Description
OB_TIME	On-board time
IP_DEADTIME_0	Dead time for Semi-module 0
IP_DEADTIME_1	Dead time for Semi-module 1
IP_DEADTIME_2	Dead time for Semi-module 2
IP_DEADTIME_3	Dead time for Semi-module 3
IP_DEADTIME_4	Dead time for Semi-module 4
IP_DEADTIME_5	Dead time for Semi-module 5
IP_DEADTIME_6	Dead time for Semi-module 6
IP_DEADTIME_7	Dead time for Semi-module 7
IP_DEADTIME_0	Dead time for Semi-module 8
IP_DEADTIME_1	Dead time for Semi-module 9
IP_DEADTIME_2	Dead time for Semi-module 10
IP_DEADTIME_3	Dead time for Semi-module 11
IP_DEADTIME_4	Dead time for Semi-module 12
IP_DEADTIME_5	Dead time for Semi-module 13
IP_DEADTIME_6	Dead time for Semi-module 14
IP_DEADTIME_7	Dead time for Semi-module 15

Table 47: Content of **COMP-DEAD-SCP** Data Structures.

Column Name	Description
OB_TIME	On-board time
IC_DEADTIME_0	Dead time for module 0
IC_DEADTIME_1	Dead time for module 1
IC_DEADTIME_2	Dead time for module 2
IC_DEADTIME_3	Dead time for module 3
IC_DEADTIME_4	Dead time for module 4
IC_DEADTIME_5	Dead time for module 5
IC_DEADTIME_6	Dead time for module 6
IC_DEADTIME_7	Dead time for module 7

C.4 *ibis_binning*

This script prepares IBIS data for scientific analysis. Its main function is to split the data into energy bins and time bins. For each time and energy ranges detector shadowgram and a corresponding efficiency shadowgram are created. Output Data Structures are listed in the Table 48. These Data Structures keep the position of each pixel and the total counts in given energy band during integration time or efficiency correspondingly. The boundaries of the energy and time bins can be found in the corresponding index Data Structures, see Table 49 for details, the abbreviations used in this table were introduced in Table 48.

C.4.1 *ii_shadow_build*

ii_shadow_build prepares ISGRI data for scientific analysis.

During the run of this executable Data Structures **ISGR-DETE-SHD** for a detector shadowgrams and **ISGR-EFFI-SHD** for ISGRI detector efficiency shadowgrams are filled. These Data Structures keep the position of each pixel and the total counts in given energy band during integration time or efficiency correspondingly.

During PICsIT analysis the same information about the PICsIT detector is written to **PICS-DETE-SHD** and **PICS-EFFI-SHD** Data Structures.

Table 48: List of Data Structures produced at BIN level

Instrument	Type of shadowgram	Output Data Structure
ISGRI	detector [ID]	ISGR-DETE-SHD
	efficiency [IE]	ISGR-EFFI-SHD
PICsIT	detector [PD]	PICS-DETE-SHD
	efficiency [PE]	PICS-EFFI-SHD

Table 49: Content of ******-****-SHD-IDX** Data Structures.

Column Name	Description	Shadowgram Type
ISDCLEVL	ISDC level of data processing	ID, IE, PD, PE
TFIRST	Time of the first data element	ID, IE, PD, PE
TLAST	Time of the last data element	ID, IE, PD, PE
TELAPSE	Total elapsed time of the data	ID, IE, PD, PE
ONTIME	Sum of good time intervals	ID, IE, PD, PE
CHANMIN	Lowest channel of the energy range	ID, IE
CHANMAX	Highest channel of the energy range	ID, IE
E_MIN	Lower bound of the energy range	ID, IE, PD, PE
E_MAX	Upper bound of the energy range	ID, IE, PD, PE
BANDTYPE	Type of energy band	ID, IE
RISE_MIN	Minimum event rise time channel 0-127	ID
RISE_MAX	Maximum event rise time channel 0-127	ID
HIS_TYPE	Type of histogram data	PD
SHD_TYPE	Shadowgram type	PE

C.5 *cat_extract*

The catalogue extraction selects the sources in the FOV from the reference catalogue. The output Data Structure **ISGR-SRCL-CAT** have the same structure as the reference catalogue **GNRL-REFR-CAT**, see Table 50.

Table 50: Content of GNRL-REFR-CAT Data Structures.

Column Name	Description
SOURCE_ID	ISDC unique source identifier
DAY_ID	Modified Julian Date of source's first identification
NAME	One commonly used name for the source
CLASS	source classification code
RA_OBJ	Source right ascension in degrees
DEC_OBJ	Source declination in degrees
ERR_RAD	Error radius
SPA_MODL	Model for source spatial extension (point, disk, ellipse, square, gaussian, Bspline, etc..)

SPA_NPAR	Number of parameters for source spatial extension
SPA_PARS	Parameters for source spatial extension
SPE_MODL	Model for source spectrum (XSPEC syntax)
SPE_NPAR	Number of parameters for source spectrum
SPE_PARS	Parameters for source spectrum
VAR_MODL	Model for source intensity variability (constant, sin, burst)
VAR_NPAR	Number of parameters for source intensity variability
VAR_PARS	Parameters for source intensity variability
COMMENTS	Comments
SPI_FLUX_1	SPI flux in the soft SPI energy band
SPI_FLUX_2	SPI flux in the hard SPI energy band
ISGR_FLUX_1	ISGRI flux in the soft ISGRI energy band
ISGR_FLUX_2	ISGRI flux in the hard ISGRI energy band
PICS_FLUX_1	PICsIT flux in the soft PICsIT energy band
PICS_FLUX_2	PICsIT flux in the hard PICsIT energy band
JEMX_FLUX_1	JEMX flux in the soft JEMX energy band
JEMX_FLUX_2	JEMX flux in the hard JEMX energy band
E_MIN	Lower energy boundaries
E_MAX	Upper energy boundaries
FLUX	Flux values
FLUX_ERR	Flux errors
SEL_FLAG	Source selection flag
FLAG	Generic flag

C.6 *ibis_background_cor*

This script combines executables performing the background correction in accordance with the chosen method.

As the output for each energy range for ISGRI shadowgrams of larger dimensions (corrected expanded shadowgram and corresponding variance and efficiency expanded shadowgrams) are written to **ISGR-CEXP-SHD**. The energy range is given in the keywords `E_MIN`, `E_MAX`, `CHAN_MIN`, `CHAN_MAX`, and a shadowgram type is given by `SHD_TYPE` keyword.

For PICsIT Data Structures **PICS-CEXP-SHD** and **PICS-VEXP-SHD** are filled for the corrected expanded detector shadowgram and the corrected expanded variance detector shadowgram. The energy range is again given in the keywords.

C.7 Image Analysis

C.7.1 *ii_skyimage*

ii_skyimage deconvolves shadowgrams in the given energy bands. The index of deconvolved and cleaned images is written to **ISGR-SKY.-IMA-IDX** Data Structure. For each output energy band defined four images are created and attached to this index.The content of this Data Structure is given in the Table 51.

Table 51: Content of **ISGR-SKY.-IMA-IDX** Data Structure.

Column Name	Description

IMATYPE	Type of image. Possible values are: "IMAGE" - deconvolved and cleaned sky image "VARIANCE" - variance image "SIGNIFICANCE" - significance map "RESIDUAL" - difference between raw (deconvolved only) and cleaned image. "EXPOSURE" - true exposure map can be created.
CHANMIN	Lowest channel of the energy range
CHANMAX	Highest channel of the energy range
E_MIN	Lower bound of the energy range
E_MEAN	Mean energy of the energy range
E_MAX	Upper bound of the energy range
TFIRST	Time of the first data element
TLAST	Time of the last data element
TELAPSE	Total elapsed time of the data
EXPOSURE	Effective exposure time
CRVAL1	LONG at the reference value
CRVAL2	LAT at the reference value

For each deconvolved image the list of found sources is created (Data Structure **ISGR-SKY.-RES**) and attached to the **ISGR-SKY.-RES-IDX** Data Structure. See content of these Data Structures in Tables 53, 52.

Table 52: Content of **ISGR-SKY.-RES** Data Structure.

Column Name	Description
NEW_SOURCE	New source flag (0 if old, 1 if new)
SOURCE_ID	ISDC unique source identifier
RA_OBJ	Source right ascension in degrees
DEC_OBJ	Source declination in degrees
Y_FIN	Y axis fine position of the source in pixels
Z_FIN	Z axis fine position of the source in pixels
FIN_YZ_ERR	Error of the fine position along the Y and Z axis in pixels
RA_FIN	Right Ascension of the fine position of the source
DEC_FIN	Declination of the fine position of the source
FIN_RD_ERR	Error of the fine position of the source in RA and DEC
FLUX	Flux values for a given energy band
FLUX_ERR	Flux errors for a given energy band
DETSIG	Source detection significance in ISGRI

Table 53: Content of **ISGR-SKY.-RES-IDX** Data Structure.

Column Name	Description
TFIRST	Time of the first data element
TLAST	Time of the last data element
TELAPSE	Total elapsed time of the data
E_MIN	Lower bound of the energy range
E_MAX	Upper bound of the energy range

Search for the sources in the FOV is then performed and a list of sources found is created (Data Structure **ISGR-MOSA-RES** has the same structure as the **ISGR-SKY.-RES**, Table52). The index of lists of sources found in each map can be found in Data Structure **ISGR-MOSA-RES-IDX** (structure analogous to the one described in the Table 53).

The output catalog containing the sources description is written to the data structure **ISGR-SRCL-RES**. This Data Structure has structure similar to input catalog **ISGR-SRCL-CAT**. Each row of this Data Structure contains the description of a source in the OG FOV. If it was present in the input catalog all the input info is copied. For each source found during the analysis the new info is added, see Table 54

The DETSIG in ISGR-SRCL-RES is calculated from the results of the mosaic image and the results of the analysis at Science Window level

$$detsig = \sqrt{\Sigma detsig_i^2},$$

where i is for all energy bands and all Science Windows and mosaic image

For the FLUX and err_flux of ISGR-SRCL-RES you have:

$$flux = \frac{\Sigma flux_i}{N}$$

$$fluxerr = \frac{\Sigma fluxerr_i}{N}$$

where i - for all Science Windows and a mosaic image for a given energy band

Table 54: New information added to the **ISGR-SRCL-RES** Data Structure.

Column Name	Description
RA_FIN	Right Ascension of the fine position of the source
DEC_FIN	Declination of the fine position of the source
FIN_RD_ERR	Error of the fine position of the source in RA and DEC
SCW_NUM_C	number of times that the source was in ScW FOV
	If SCW_NUM_C = 0 then it is a new source
SCW_NUM_F	number of times that the source was found in ScW image
OG_NUM	number of times that the source was found in OG image

With the help of stand alone program *src_collect* it si possible to collect results from different science windows. Results are written to the **ISGR-OBS.-RES** Data Structure, see Table 55.

Table 55: Content of the **ISGR-OBS.-RES** Data Structure.

Column Name	Description
SWID	Science Window identifier
SOURCE_ID	ISDC unique source identifier
NAME	One commonly used name for the source
RA_FIN	Source right ascension in degrees
DEC_FIN	Source declination in degrees
FIN_RD_ERR	Error of the fine position of the source in RA and DEC
Y_FIN	Y axis fine position of the source in pixels
Z_FIN	Z axis fine position of the source in pixels
FIN_YZ_ERR	Error of the fine position along Y and Z axis in pixels
DETSIG	Source detection significance in ISGRI
E_MIN	Lower energy boundaries
E_MAX	Upper energy boundaries
FLUX	Flux values
FLUX_ERR	Flux errors
DEADC	Mean deadtime (and greyfilter) correction factor
EXPOSURE	Mean exposure time over the detector plane
TSTART	Start time of the observation (IJD)

TSTOP	End time of the observation (IJD)

C.7.2 *ip_skyimage*

ip_skyimage performs deconvolution of the PICsIT shadowgrams with the use of the balanced cross-correlation method described in Goldwurm et al. 2003 [10], see also Section 12.8. The values of the decoding G-array are taken from **PICS-DECO-MOD** Data Structure.

The output Data Structure **PICS-SKY.-IMA** store either a cartesian (CAR) or a tangential (TAN) projection of the celestial sphere. The index of deconvolved images is written to the **PICS-SKY.-IMA-IDX** Data Structure, it has the same content as **ISGR-SKY.-IMA-IDX** (Table 51). IMATYPE can be either "IMAGE", either "VARIANCE", or "SIGNIFICANCE".

The list of sources that were found during the analysis are written to the **PICS-SKY.-RES**

C.8 Spectral Analysis

C.8.1 *ii_spectra_extract*

ii_spectra_extract reads the shadowgram for each input energy band and creates a spectrum for background and each source from isgri_srcl_res.fits file. The result is written to the ISGR-EVTS-SPE and ISGR-EVTS-SPE-IDX Data Structures.

Table 56: Content of the **ISGR-EVTS-SPE** Data Structure.

Column Name	Description
CHANNEL	Channel number
RATE	Countrate in the given channel
SYS_ERR	Systematic error
STAT_ERR	Statistical error
QUALITY	Quality flag
GROUPING	Grouping flag

Table 57: Content of the **ISGR-EVTS-SPE-IDX** Data Structure.

Column Name	Description
SOURCEID	ISDC unique source identifier
RA_OBJ	Source right ascension in degrees
DEC_OBJ	Source declination in degrees
TFIRST	Time of the first data element
TLAST	Time of the last data element
TELAPSE	Total elapsed time of the data

C.9 Timing Analysis

C.9.1 *ip_st_lc_extract*

This program build light curves and related errors starting from spectral timing data of PICsIT. The content of the filled Data Structures is given in the Tables 58, 59.

Table 58: Content of the **PICS-EVTS-LCR-IDX** Data Structure.

Column Name	Description
SOURCEID	ISDC unique source identifier
CHANMIN	Lowest channel of the energy range
CHANMAX	Highest channel of the energy range
E_MIN	Lower bound of the energy range
E_MAX	Upper bound of the energy range
PICSMODE	Mode of the PICsIT detector layer

Table 59: Content of the **PICS-EVTS-LCR** Data Structure.

Column Name	Description
TIME	Time of measurement for the bin
TOT_COUNTS	Total counts of the source region
BACKV	Background counts scaled to the source region
BACKE	Background count errors
ERROR	Net count error in the source region
RATE	Countrate in the given energy band
FRACEXP	Fraction of integration bin time for exposure correction

C.9.2 *ii_lc_extract*

For all sources from the isgri_srcl_res.fits file, *ii_lc_extract* extracts the ISGRI lightcurves. The results are written to the ISGR-SRC.-LCR and ISGR-SRC.-LCR-IDX Data Structures. ISGR-SRC.-LCR has the same structure as PICS-EVTS-LCR, see Table 59

Table 60: Content of the **ISGR-SRC.-LCR-IDX** Data Structure.

Column Name	Description
SOURCEID	ISDC unique source identifier
RA_OBJ	Source right ascension in degrees
DEC_OBJ	Source declination in degrees
CHANMIN	Lowest channel of the energy range
CHANMAX	Highest channel of the energy range
E_MIN	Lower bound of the energy range
E_MAX	Upper bound of the energy range

C.9.3 Timing Analysis without the deconvolution

During the run of stand alone program*evts_extract* two data structures **GNRL-EVTS-GTI** and **GNRL-EVTS-LST** are filled.

GNRL-EVTS-LST (Table 61) combines all available information for photon-by-photon events from different instruments during a given time interval.

Table 61: Content of the **GNRL-EVTS-LST** Data Structure.

Column Name	Description
DETY	Y location in the detector layer (offset from center)
DETZ	Z location in the detector layer (offset from center)
ENERGY	Energy deposited by the event
EVNT_TYPE	Type and origin of event (bit coded)
TIME	Time of event in INTEGRAL Julian Date units

TIMEDEL	Uncertainty of time stamp
DEADC	Dead time correction factor
BARYTIME_N	Barycenter time for source number N
PIF_N	Pixel Illumination Factor for source number N
AREASCAL_N	Nominal effective area for source number N

GNRL-EVTS-GTI (Table 62) contains good time intervals for selecting events.

Table 62: Content of the **GNRL-EVTS-GTI** Data Structure.

Column Name	Description
START	IJD of start of the GTI
STOP	IJD of end of the GTI
OBT_START	On-board time of start of the GTI
OBT_END	On-board time of end of the GTI
UTC_START	UTC of start of the GTI
UTC_END	UTC of end of the GTI

D List of *ibis_science_analysis* parameters

Table 63: *ibis_science_analysis* parameters description. Query parameters are marked with bold font.

Name	Type	Description
General parameters		
ogDOL	string	DOL of the Observational Group to be analyzed. default: ``og_ibis.fits''
startLevel	string	Analysis level at which the analysis begins. The names of the possible analysis levels are given in section 5. Possible values: ``COR'' - ``COMP''. Also ``CLEAN'' value is possible. default: ``COR''
endLevel	string	Analysis level at which analysis finishes. The names of the possible analysis levels are given in the section 5. possible values: COR - CLEAN. **We recommend you to proceed by steps, as described in the cookbook.** default: "IMA2"
staring	boolean	Enter yes if it was a staring observation. It has an influence on PICsIT analysis - for staring observation all shadowgrams are summed before the deconvolution. default: no
tolerance	real	tolerance for staring. In case of **staring=**yes the check is performed that during the Science Windows within the Observation Group the spacecraft was stable within the given tolerance. default: 0.1
sum_spectra	boolean	Do the summing of spectra? default: "no"
CAT_refCat	string	DOL of Reference Catalog default: ``$ISDC_REF_CAT[ISGRI_FLAG>0]''
CAT_usrCat	string	DOL of User Catalog (**CURRENTLY IGNORED!!!**)
chatter	integer	Verbosity level possible values: 0 - 5, 0 = errors only, 1 = warnings, 2 = normal, ... default: 2
IC_Group	string	DOL of Instrument Characteristics master group default: ``$REP_BASE_PROD/idx/ic/ic_master_file.fits''
IC_Alias	string	Selection alias for Instrument Characteristics default: ``OSA''

corrDol	string	DOL of the isgri off-axis corrections default: “ ”
rebinned_corrDol_ima	string	DOL of the rebinned isgri off-axis corrections for imaging default: “ ”
rebinned_corrDol_spe	string	DOL of the rebinned isgri off-axis corrections for spectrum default: “ ”
rebinned_corrDol_lc	string	DOL of the rebinned isgri off-axis corrections for lc default: “ ”
rebinned_backDol_ima	string	DOL of the rebinned isgri back corrections for imaging default: “ ”
rebinned_backDol_spe	string	DOL of the rebinned isgri back corrections for spectrum default: “ ”
rebinned_backDol_lc	string	DOL of the rebinned isgri back corrections for lc default: “ ”
GENERAL_clobber	boolean	Clobber existing output data? default: `YES`
GENERAL_levelList	boolean	List of all levels default: `COR,GTI,DEAD,BIN_I,BKG_I,CAT_I,IMA,IMA2,BIN_S,SPE,LCR,COMP,CLEAN`
Parameters to define which part of data should be analyzed		
SWITCH_disableIsgri	boolean	Disable ISGRI analysis? possible values: `YES, NO` default: `NO`
SWITCH_disablePICsIT	boolean	Disable PICsIT analysis? possible values: `YES, NO` default: `YES`
SWITCH_disableCompton	boolean	Disable Compton analysis? possible values: `YES, NO` default: `YES`
SWITCH_osimData	boolean	Data are from simulator? possible values: `YES, NO` default: `NO`
Parameters for ibis binning and ibis image reconstruction		
IBIS_II_ChanNum	integer	Number of output energy bands for ISGRI possible values: `-1` – `10` value `-1` is for more than 10 energy bands, in this case `IBIS_II_inEnergyValues` should be set. default: `4`

IBIS_II_E_band_min	string	List of lower limits of output energy bands (keV) for ISGRI default: ``20 40 60 100''
IBIS_II_E_band_max	string	List of upper limits of output energy bands (keV) for ISGRI default: ``40 60 100 200''
IBIS_II_inEnergyValues	string	DOL of the energy values when required. default: `` ''
IBIS_IPS_corrPDH	integer	Packets threshold for partially downloaded histograms. default:0
IBIS_IPS_ChanNum	integer	Number of Energy bands for PICsIT in standard mode. possible values: 0 – 256 You should set it to -1 to build automatically the only binning for which background maps are present. It corresponds to the following energy ranges (keV): 203 –252, 252 – 329, 329 – 455, 455 – 655, 655 – 1057, 1057 – 1841, 1841 – 3570, 3570 – 6510. default: -1
IBIS_IPS_E_band_min_s	string	List of lower energy boundary for PICsIT in standard mode for SINGLE. **Do not touch this parameter!** default: ""
IBIS_IPS_E_band_max_s	string	List of higher energy boundary for PICsIT in standard mode for SINGLE. **Do not touch this parameter!** default: “”
IBIS_IPS_E_band_min_m	string	List of lower energy boundary for PICsIT in standard mode for MULTIPLE. **Do not touch this parameter!** default: “”
IBIS_IPS_E_band_max_m	string	List of higher energy boundary for PICsIT in standard mode for MULTIPLE. **Do not touch this parameter!** default: “”
IBIS_IP_ChanNum	integer	Number of Energy bands for PICsIT in photon-by-photon mode. possible values: 1 – 300 **Do not touch this parameter!** default: 3
IBIS_IP_E_band_min_s	string	List of lower energy boundary (single) **Do not touch this parameter!** default (keV): ``175 600 1000''

IBIS_IP_E_band_max_s	string	List of higher energy boundary (single) **Do not touch this parameter!** default (keV): ``600 1000 10000''
IBIS_IP_E_band_min_m	string	List of lower energy boundary (multiple) **Do not touch this parameter!** default (keV): ``350 600 1000''
IBIS_IP_E_band_max_m	string	List of higher energy boundary (multiple) **Do not touch this parameter!** default (keV): ``600 1000 13500''
IBIS_NoisyDetMethod	integer	Noisy Pixel detection method possible values: 0 (only normal noisy detection), 1 (add also spectral noisy detection) default: 1
These parameters are recommended by the IBIS team, better not to change them.		
IBIS_min_rise	integer	Minimum rise time. default: 16
IBIS_max_rise	integer	Maximum rise time. default:116
IBIS_P_convFact	real	Conversion factor channel-to-energy. default: 7.1
ModPixShad	integer	Minimum number of non-illuminated pixels per module possible values: 100 - 500 default:400
Parameters specific to SCW1 pipeline.		
SCW1_ISGRI_event_select	string	CFITSIO event selection string default: `` ''
SCW1_GTI_LimitTable	string	DOL of the limit table for GTIs default: `` ''
SCW1_GTI_attTolerance_X	real	Accepted attitude stability tolerance of X (optical) axis to generate a GTI [arc min] possible values: 0 – 10800 default: 0.5
SCW1_GTI_attTolerance_Z	real	Accepted attitude stability tolerance of Z axis to generate a GTI [arc min]" possible values: 0 – 10800 default: 3.0
SCW1_GTI_gtiUserP	string	DOL of the user GTI table for PICsIT default: `` ''
SCW1_GTI_gtiUserI	string	DOL of the user GTI table for ISGRI default: `` ''

SCW1_GTI_TimeFormat	string	Time format to be used possible values: ``‘‘IJD’’``, ``‘‘UTC’’``, ``‘‘OBT’’`` default: ``‘‘IJD’’``
SCW1_GTI_Accuracy	string	Used accuracy for OBT to IJD conversion and vice versa possible values: ``‘‘any’’``, ``‘‘inaccurate’’``, ``‘‘accurate’’`` default: ``‘‘any’’``
SCW1_GTI_SCP	string	Names of spacecraft GTIs to be merged for PICsIT default: ``‘‘ ’’``
SCW1_GTI_SCI	string	Names of spacecraft GTIs to be merged for ISGRI default: ``‘‘ ’’``
SCW1_GTI_PICsIT	string	GTIs to be merged for PICsIT default: ``‘‘VETO ATTITUDE P_SGLE_DATA_GAPS P_MULE_DATA_GAPS’’``
SCW1_GTI_ISGRI	string	GTIs to be merged for ISGRI default: ``‘‘VETO ATTITUDE ISGRI_DATA_GAPS’’``
SCW1_GTI_BTI_Dol	string	DOL of a bad time interval table (GNRL-INTL-BTI) default: ``‘‘IBIS_CONFIGURATION IBIS_BOOT ISGRI_RISE_TIME VETO_PROBLEM SOLAR_FLARE BELT_CROSSING’’``
SCW1_GTI_BTI_Names	string	Input BTI names to be considered default: ``‘‘ ’’``
SCW1_ICOR_idxSwitch	string	DOL of the index of pixels switches list. default: ``‘‘ ’’``
SCW1_ICOR_GODOL	string	DOL of the gain-offset table default: ``‘‘ ’’``
SCW1_ICOR_supGDOL	string	DOL of gain coefficients for 2nd method default: ``‘‘ ’’``
SCW1_ICOR_supODOL	string	DOL of offset coefficients for 2nd method” default: ``‘‘ ’’``
SCW1_ICOR_riseDOL	string	DOL of the rise time correction table default: ``‘‘ ’’``
switDOL	string	switch on time default: ``‘‘ ’’``
SCW1_ICOR_rtcDOL	string	rise time correction default: ``‘‘ ’’``

SCW1_RTdriftCor	integer	RT correction level increasing from 0 to 2.If 0, no RT correction, 2 RT drift included into corrections, 1 default value default: `1`
SCW1_ICOR_probShot	real	Probability of shot time decay. default: `0.0001`
SCW1_PCOR_enerDOL	string	DOL of the energy correction table (PICsIT) default: `‘‘ ’’`
SCW1_BIN_cleanTrk	integer	Cleaning of cosmic-ray induced events. possible values: `0` = No `1` = Yes default: `1`
SCW1_veto_mod	string	DOL of the IC file for VETO model and width of Compton window default: `‘‘ ’’`
SCW1_BIN_I_idxNoisy	string	DOL of the index of noisy maps. default: `‘‘ ’’`
SCW1_BIN_I_idxLowThre	string	index of Low Threshold default: `‘‘ ’’`
SCW1_BIN_P_inDead	string	DOL of the dead time data structure. default: `‘‘ ’’`
SCW1_BIN_P_inGTI	string	DOL of the good time interval data structure. default: `‘‘ ’’`
SCW1_BIN_P_PicsCxt	string	DOL of the PICsIT Context Tables. default: `‘‘ ’’`
SCW1_BIN_P_HepiLut	string	DOL of the hepi lut default: `‘‘ ’’`
SCW1_BKG_divide	boolean	Divide by efficiency? **Do not touch it!** default: `no`
SCW1_BKG_badpix	boolean	Remove bad pixels? **Do not touch it!** default: `yes`
SCW1_BKG_flatmodule	boolean	flatten modules? **Do not touch it!** default: `no`
SCW1_BKG_I_isgrUnifDol	string	DOL of the ISGRI detector uniformity possible values: DOL for a specific model `‘‘-’’` – no uniformity correction `‘‘ ’’` – DOL is taken from the IC tree default: `‘‘-’’`

SCW1_BKG_I_isgrBkgDol	string	DOL of the ISGRI background model possible values: DOL for a specific model `‘‘-’’` – no background subtraction `‘‘ ’’` – DOL is taken from the IC tree default: `‘‘ ’’`
SCW1_BKG_I_method_cor	integer	Method to be applied for background removal possible values: `0` – no background correction `1` – background from IC tree is applied to the whole detector `2` – background is threaded for each module separately When using 1 or 2, `brPif` and `brPifThreshold` parameters are taken into account. default: `‘‘1’’`
brSrcDOL	string	DOL of the bright sources catalogue, which will be removed from background mean calculation default: `‘‘$ISDC_REF_CAT[ISGRI_FLAG>0&&ISGR_FLUX_1>100]’’`
SCW1_BKG_I_brPifThreshold	real	pixels with PIF value higher will be removed from background calculation. When = 1, bright PIF removal will not be performed possible values: `‘‘0 -- 1 ’’` default: `‘‘0.0001’’`
SCW1_BKG_I_pif	string	filename of the pif in every ScW default: `‘‘ ’’`
SCW1_BKG_I_method_int	integer	Method to be applied for the pixel value interpolation possible values: `0` – pixels in dead zones are filled with zeros `1` – pixels are filled with mean detector value default: `‘‘1’’`
tungAtt	string	DOL of the Tungsten attenuation length fits file default: `‘‘ ’’`
aluAtt	string	DOL of the Aluminium attenuation length fits file default: `‘‘ ’’`
leadAtt	string	DOL of the Lead attenuation length fits file default: `‘‘ ’’`
SCW1_BKG_P_method	integer	Method applied in filling gaps possible values: `0` – model (in cts) is normalized by the time of observation `1` – model (in cts) is normalized by the mean count value default: `0`
SCW1_BKG_picsSUnifDOL	string	DOL of the PICsIT Detector Uniformity model (single). default: `‘‘-’’`

SCW1_BKG_picsSBkgDOL	string	DOL of the PICsIT Detector Background model (single). default: ‘‘ ’’
SCW1_BKG_picsMUnifDOL	string	DOL of the PICsIT Detector Uniformity model (multiple). default: ‘‘-’’
SCW1_BKG_picsMBkgDOL	string	DOL of the PICsIT Detector Background model (multiple). default: ‘‘ ’’
Parameters specific to OBS1 pipeline.		
`ii_skyimage` parameters		
OBS1_DataMode	integer	Data Simulation mode possible values: `0` – shadowgrams are treated `1` – shadowgrams are simulated default: `0`
rebin_slope	real	Spectral slope to be used default: `-2`
rebin_arfDol	string	DOL of the ISGRI arf for rebinning. default: ‘‘ ’’
OBS1_SearchMode	integer	Source search mode possible values for the Science Window analysis: `1` – search for all catalog sources `2` – search for K sources,where K= **ToSearch** `3` – search for all catalog sources plus for K significant excesses,where K = **ToSearch** For *all* the above, when the mosaic is created, the software looks for K = **ToSearch** sources. default: `3`
OBS1_ToSearch	integer	Number of sources to be looked for default: `50`
OBS1_CleanMode	integer	Ghost cleaning possible values: `1` – source model lobes are subtracted from the deconvolved image `-1` – no subtraction default: `1`
OBS1_ExtenType	integer	Save exposure/residual maps possible values: `absent`, `0` – `3`, `0` or `absent` - no action `1` - true exposure maps are saved in 4*n extension of isgri_sky_ima.fits instead of residual images `2` - true exposure maps are saved in the 5*n extension of isgri_sky_ima.fits after the residual images `3` - one true exposure map is created for the last energy band and saved in the last extension of isgri_sky_ima.fits default: `2`

OBS1_NegModels	integer	0 or absent for no negative models ; 1 - negative models default: `0`
OBS1_FastOpen	integer	if 1 then no CommonPreparePars default: `1`
ISGRI_mask	string	DOL of the MASK pattern fits file default: `‘‘ ’’`
OBS1_deco	string	DOL of the projected decoding pattern fits file default: `‘‘ ’’`
OBS1_covrMod	string	DOL of the covariance fits file default: `‘‘’’`
OBS1_ScwType	string	Type of Science Window to be treated possible values: `‘‘POINTING’’`, `‘‘SLEW’’`, `‘‘OTHER’’`, `‘‘ANY’’` default: `‘‘POINTING’’`
OBS1_DoPart2	integer	if `1` or absent → do mosaic part. Set it to `0` if don’t want to produce a mosaic image. Set it to `2` if you want to produce a mosaic image from different existing runs. default: `1`
OBS1_MapAlpha	real	Mosaic map center [deg] default: `""`
OBS1_MapDelta	real	Mosaic map center [deg] default: `""`
OBS1_MapSize	real	Mosaic map radius [deg] default: `40.0`
OBS1_PixSpread	integer	`0` – no flux spread in mosaic default: `1`
OBS1_MinCatSouSnr	real	The software detects a catalog source only if its signal-to-noise ratio is higher than this value. default: `6`
OBS1_MinNewSouSnr	real	The software detects a catalog source only if its signal-to-noise ratio is higher than this value. default: `7`
OBS1_SouFit	integer	defines whether to fit or not the source position `0` – for fitting source position in ScW fit `1` – for fixed source position in ScW fit default: `0`
PICSIT_detThr	real	Detection Threshold default: `3.0`
PICSIT_inCorVar	integer	Corrected variance to be used [0=NO; 1=YES] default: `1`

PICSIT_outVarian	integer	Variance maps in output [0=NO; 1=YES] default: `0`
cat_extract parameters, see Section 12.7 for more details.		
OBS1_CAT_radiusMin	string	Low limit for the position selection. default: ``0''
OBS1_CAT_radiusMax	string	High limit for the position selection. default: ``20''
OBS1_CAT_fluxDef	string	Column used for flux selection. default: `` ''
OBS1_CAT_fluxMin	string	Low limit for flux selection. default: `` ''
OBS1_CAT_fluxMax	string	High limit for flux selection. default: `` ''
OBS1_CAT_class	string	Select the object by class. default: `` ''
OBS1_CAT_date	real	IJD for the public data. default: −1
Parameters specific to SCW2 pipeline.		
PICSIT_deco	string	DOL of the MASK decoding fits file default: `` ''
SCW2_cat_for_extract	string	Catalog for spectral and lightcurve extraction (if empty then it is taken from the imaging result of the Science Window) default: `` ''
SCW2_racolumn	string	Name of the column where to get Ra default: ``RA_FIN''
SCW2_deccolumn	string	Name of the column where to get Dec default: ``DEC_FIN''
SCW2_catalog	string	Catalog for PICsIT imaging default: `` ''
SCW2_PIF_filter	string	filter to apply on the source default: `` ''
SCW2_ISGRI_event_select	string	CFITSIO event selection string default: `` ''
parameter for building light curves from PICsIT spectral timing data		
Parameters for spectra extraction.		
IBIS_SI_ChanNum	integer	Number of Channels possible values: `-1` – `10`, `-1` = take from **ISGR-EBDS-MOD** structure default: `-1`

IBIS_SI_E_band_min	string	List of lower limits of output energy bands default: ‘‘ ’’
IBIS_SI_E_band_max	string	List of upper limits of output energy bands default: ‘‘ ’’
IBIS_SI_inEnergyValues	string	DOL of the energy values when required. default: ‘‘ ’’
PICSIT_source_name	string	Name of the source for extracting PICsIT spectrum default: ‘‘ ’’
PICSIT_source_RA	real	RA of the source for extracting PICsIT spectrum default: ‘‘ ’’
PICSIT_source_DEC	real	DEC of the source for extracting PICsIT spectrum default: ‘‘ ’’
IBIS_SS_inEnergyValues	string	DOL of the energy values for single events default: ‘‘ ’’
IBIS_SM_inEnergyValues	string	DOL of the energy values for multiple events default: ‘‘ ’’
IBIS_SP_ChanNum	integer	Number of Channels possible values: `0 – 300` default: `51`
IBIS_SP_E_band_min_s	string	List of lower limits of output energy bands (single) default: ‘‘ ’’
IBIS_SP_E_band_max_s	string	List of upper limits of output energy bands (single) default: ‘‘ ’’
IBIS_SP_E_band_min_m	string	List of lower limits of output energy bands (multiple) default: ‘‘ ’’
IBIS_SP_E_band_max_m	string	List of upper limits of output energy bands (multiple) default: ‘‘ ’’
IBIS_SPS_ChanNum	integer	Number of Channels possible values: `-1 – 256` default: `-1`
IBIS_SPS_E_band_min_s	string	List of lower limits of output energy bands for SINGLE default: ‘‘ ’’
IBIS_SPS_E_band_max_s	string	List of upper limits of output energy bands for SINGLE default: ‘‘ ’’
IBIS_SPS_E_band_min_m	string	List of lower limits of output energy bands for multiple default: ‘‘ ’’
IBIS_SPS_E_band_max_m	string	List of upper limits of output energy bands for multiple default: ‘‘ ’’

SCW2_ISPE_idx_isgrResp	string	DOL of the index of ISGRI response matrices default: `''`
SCW2_ISPE_isgrarf	string	DOL of the ARFs for XSPEC default: `''`
SCW2_ISPE_DataMode	integer	Data Simulation mode default: `0`
SCW2_ISPE_MethodInt	integer	Method to be applied for the pixel value interpolation default: `1`
SCW2_ISPE_MethodFit	integer	Method to be applied for background and source intensity fitting default: `6`
SCW2_ISPE_isgrUnifDol	string	DOL of the ISGRI detector uniformity default: `''`
SCW2_BIN_cleanTrk	integer	Cleaning of cosmic-ray induced events. possible values: `0` = No `1` = Yes default: `0`
SCW2_BIN_I_idxNoisy	string	DOL of the index of noisy maps. default: `''`
SCW2_BIN_I_idxLowThre	string	index of Low Threshold default: `''`
SCW2_BIN_P_inDead	string	DOL of the dead time data structure. default: `''`
SCW2_BIN_P_inGTI	string	DOL of the good time interval data structure. default: `''`
SCW2_BIN_P_PicsCxt	string	DOL of the PICsIT Context Tables. default: `' '`
SCW2_BIN_P_HepiLut	string	DOL of the hepi lut default: `''`
SCW2_BKG_I_isgrBkgDol	string	DOL of the isgri bckg model or - if you want none or empty if you want DOL automatic from the IC tree default: `''`
SCW2_BKG_divide	boolean	Divide by efficiency default: `no`
SCW2_BKG_badpix	boolean	Remove bad pixels default: `yes`
SCW2_BKG_flatmodule	boolean	flatten modules default: `no`

SCW2_BKG_I_method_cor	string	Method to be applied for background removal possible values: `0` – background from IC tree is applied to the whole detector `1` – parameter switches to this value automatically when background map is specified (not an IC one) `2` – background is threaded for each module separately When using 1 or 2, `brPif` and `brPifThreshold` parameters are taken into account. default: ``‘‘2’’``
SCW2_BKG_I_method_int	string	Method to be applied for the pixel value interpolation possible values: `0` – pixels in dead zones are filled with zeros `1` – pixels are filled with mean detector value default: ``‘‘1’’``
SCW2_BKG_P_method	integer	Method applied in filling gaps possible values: `0` – `2` default: `1`
SCW2_BKG_picsSUnifDOL	string	DOL of the PICsIT Detector Uniformity model (single). default: ``‘‘’’``
SCW2_BKG_picsSBkgDOL	string	DOL of the PICsIT Detector Background model (single). default: ``‘‘-’’``
SCW2_BKG_picsMUnifDOL	string	DOL of the PICsIT Detector Uniformity model (multiple). default: ``‘‘’’``
SCW2_BKG_picsMBkgDOL	string	DOL of the PICsIT Detector Background model (multiple). default: ``‘‘-’’``
Parameters for lightcurve extraction.		
ILCR_select	string	Event selection condition default: ``‘‘’’``
ILCR_delta_t	real	Time bin in seconds. possible values: `0.1` – `10000` default: `100`
ILCR_num_e	integer	Number of energy channels possible values: `1` – `10` default: `4`
ILCR_e_min	string	List of low energy boundaries default: “`20 40 60 100`”
ILCR_e_max	string	List of high energy boundaries default: “`40 60 100 200`”
Parameters for PICsIT mosaic extraction.		

OBS2_detThr	real	Detection Threshold (in sigmas). default: `3.0`
OBS2_projSel	string	Selection of projection default: `STG`
OBS2_imgSel	string	Selection criteria of images. default: `"EVT_TYPE=='SINGLE' && E_MIN==252 && E_MAX==336"`

References

[1] Introduction to the *INTEGRAL* Data Analysis.
http://www.isdc.unige.ch/integral/analysis#Documentation

[2] IBIS Observer's Manual
http://integral.esac.esa.int/AO10/AO10_IBIS_ObsMan.pdf

[3] ISGRI Scientific Validation Report
http://www.isdc.unige.ch/integral/analysis#Documentation

[4] PICsIT Analysis Scientific Validation Report
http://www.isdc.unige.ch/integral/analysis#Documentation

[5] *INTEGRAL* Off-line Scientific Analysis Installation Guide
http://www.isdc.unige.ch/integral/download/osa/doc/current/osa_inst_guide.pdf

[6] Di Cocco G., Caroli E., Celesti E., et al., 2003, A&A 411, L189

[7] A. Goldwurm, 1995, Exper.Astron., 6, 9

[8] A. Goldwurm et al.,2000, AIP, 510, 703

[9] A. Goldwurm et al., 2001, Proc. of 4th *INTEGRAL* Workshop, ESA SP 459, p. 497

[10] Goldwurm A., David P., Foschini L., et al., 2003, A&A 411, L223

[11] S.R. Gottesman & E.F. Fenimore, 1989, Applied Optics 28, 4344

[12] G.K. Skinner, R.M. Rideout, 1995, Imaging in High Energy Astronomy: 177-182

[13] Walter, R.; Favre, P.; Dubath, P. et al., 2003, A&A 411, L25

[14] Lebrun et al. 2012 in preparation

[15] Grinberg, V. et al. 2011, AcPol 51b, 33